Ultra-Low Loss Optical Fiber Characterization System Development

Final Report for Naval Research Laboratory
Project Number 65-9017-88

1 June 1990

VPI&SU Project No. 230-11 - 110f-104-352298
U.S. SFRC No. N00014-88-K-2013

Principal Investigator: Dr. R.O. Claus

Fiber & Electro-Optics Research Center
Bradley Department of Electrical Engineering
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061-0111
1.0 Introduction .............................................. 3
1.1 Measurements Performed by the NRL IR System 1 .... 3
1.1.1 Spectral Attenuation ................................ 3
1.1.2 Differential Modal Attenuation ..................... 3
1.1.3 Numerical Aperture .................................... 4
1.2 System Specifications ..................................... 4
1.2.1 Fiber Limitations ....................................... 4
1.2.2 Detector Noise (RMS values) ....................... 4
1.2.3 Spectral Signal-to-Noise .............................. 4
1.2.4 Lamp Drift ............................................. 5

2.0 How the NRL IR System 1 differs from the Photon Kinetics FOA 2000 ........................................... 6
2.1 Monochromator ............................................. 6
2.2 Infrared Lenses ............................................ 7
2.3 Quartz-Tungsten Lamp .................................... 7
2.4 Cutoff Filters ............................................. 7
2.5 Infrared Neutral Density Filters ....................... 8
2.6 Indium Antimonide Detector ............................ 8
2.7 Vacuum Chucks ........................................... 8
2.8 Annular Apertures for DMA Measurements ............ 8
2.9 Knife Edge Numerical Aperture Measurements ...... 9
2.10 Major Changes in the FOA-2000 Software .......... 9

3.0 Normal Operating Procedures ................................ 13
3.1 Turn-on Procedure ....................................... 13
3.2 Software Set-up .......................................... 13
3.3 Fiber End Preparation .................................... 14
3.4 Important Commands ..................................... 14

4.0 Maintenance ............................................... 16
4.1 Alignment Procedure for NRL IR System 1 .......... 16
4.1.1 Definitions ........................................... 16
4.1.2 Main LED Beam Path ................................ 16
4.1.3 Input Objective Lens ................................ 17
4.1.4 Input Fiber Chuck .................................... 17
4.1.5 Launch Spot .......................................... 18
4.1.6 Output Objective Lens ................................ 18
4.1.7 Output Fiber Chuck .................................. 18
4.1.8 Output Target ........................................ 19
4.1.9 Lamp Path ............................................. 19
4.1.10 Germanium Detector ................................ 19
4.1.11 Indium Antimonide Detector ...................... 20

5.0 How to get help ............................................ 21
5.1 FOA-2000 Error Messages ................................ 21
5.2 Likely Problems .......................................... 22

Appendix A. Neutral Density Filter Calibration Results .... 24
Appendix B. InSb Detector Preamp Power Supply Circuit .... 25
Appendix C. Index of Technical Reports and Publications 26
Appendix D. System Software Listing ........................ 27
1.0 Introduction

The NRL IR System 1 is an automated optical bench designed for the measurement of spectral attenuation, differential modal attenuation, and numerical aperture of zirconium fluoride infrared optical fiber. It was developed by the Fiber & Electro-Optics Research Center at Virginia Tech under contract to NRL, and is a specially adapted version of a commercially available FOA-2000, a silica fiber characterization system manufactured by Photon Kinetics of Beaverton, Oregon.

1.1 Measurements performed by the NRL IR System 1

1.1.1 Spectral Attenuation

The system measures spectral attenuation over the range from 600 nm to 4 μm. Launch conditions are overfilled for multimode fibers with core diameters up to 150 μm and with numerical apertures up to 0.24. The fiber vacuum chucks can accept fibers with outside diameters up to 200 μm. The attenuation is derived by performing a cut-back test.

1.1.2 Differential Modal Attenuation

The system can measure differential modal attenuation (DMA) on step-index multimode fibers. Launch conditions are restricted by manually placing apertures that restrict the launch numerical aperture (NA) to a narrow range, thereby exciting a limited mode group in the optical fiber. Attenuation values are derived by performing a cutback test. The DMA apertures and the range of NA for each aperture follows:

<table>
<thead>
<tr>
<th>Aperture #</th>
<th>NA Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>.04</td>
</tr>
<tr>
<td>#2</td>
<td>.08</td>
</tr>
<tr>
<td>#3</td>
<td>.10</td>
</tr>
<tr>
<td>#4</td>
<td>.13</td>
</tr>
<tr>
<td>#5</td>
<td>.15</td>
</tr>
<tr>
<td>#6</td>
<td>.18</td>
</tr>
<tr>
<td>#7</td>
<td>.20</td>
</tr>
<tr>
<td>#8</td>
<td>.04&lt;NA&lt;.08</td>
</tr>
<tr>
<td>#9</td>
<td>.08&lt;NA&lt;.13</td>
</tr>
<tr>
<td>#10</td>
<td>.11&lt;NA&lt;.17</td>
</tr>
<tr>
<td>#11</td>
<td>.14&lt;NA&lt;.21</td>
</tr>
</tbody>
</table>

DMA tests for graded-index fibers have not been implemented in the current version of the system software, but the system may be easily adapted for this test. In order to achieve the correct restricted launch for graded index fibers,
the launch spot size must be restricted as well as the numerical aperture. To restrict the spot size, an aperture of the correct diameter must be placed in the spot restrictor carriage holder (see Figures 1 and 2), and the DMA software must be changed to engage the spot restrictor. The spot restrictor aperture is demagnified 100/9 times when it is imaged onto the input fiber end. The current spot restrictor aperture (390 μm diameter) achieves a spot size of 35 μm on the end of the fiber. This represents the minimum spot achievable at 2.5 μm wavelength since it is approximately the diffraction limit for the infrared lenses at that wavelength.

### 1.1.3 Numerical Aperture

The system will measure the numerical aperture of step and graded-index fibers, up to a value of 0.24. Due to the low radiance of the lamp source, the measurement procedure uses a scanning knife-edge, rather than a scanning aperture in the far-field as specified in EIA FOTP #47. The knife edge technique is an adaptation of an EIA procedure for determining the mode field diameter of a single mode fiber. In this procedure, a knife edge is scanned across the far field output of the fiber, and a lens is used to collect the light passed by the knife edge and direct it to the detector. The computer reads the output of the detector at the lock-in amplifier, which is effectively the integrated power as a function of far field angle. This data is then differentiated and smoothed to yield the far field radiation pattern of the fiber. From this far field the numerical aperture is derived. A criterion of 5% of maximum intensity is used to determine the numerical aperture.

### 1.2 System specifications

#### 1.2.1. Fiber limitations

The system provides overfilled launch conditions for multimode fibers with core diameters up to 150 μm and numerical apertures up to 0.24. The differential modal attenuation procedure is currently set up for step index fibers only.

#### 1.2.2. Detector noise (RMS values)

The following values for the average detector noise were measured using the “SUB low_init_check” subroutine in the FOA-2000 QC software package.

- Thermoelectrically cooled Germanium detector: 0.66 μV
- Liquid nitrogen cooled Indium Antimonide detector: 0.7 μV

#### 1.2.3. Spectral signal-to-noise

After the values for detector noise given above were determined, the FOA-2000 QC software was used to measure the spectral signal-to-noise by
running the "SUB Spec\_snoise" subroutine. A one-meter piece of fluoride fiber (from NRL spool number 891019) was used to give a representative value for coupling losses into a fluoride fiber. The results are graphed below.

![Graph 1. Spectral signal-to-noise ratio](image)

1.2.4 Lamp drift

The FOA-2000 QC software subroutine titled "SUB Lamp\_drift" was used to measure the drift of the lamp output as detected by both the germanium (Ge) detector and the indium antimonide (InSb) detector. This test measured the detector output every twenty seconds over a duration of 60 minutes. The test indicated that the germanium detector drifted by -0.002 dB per five minutes. The indium antimonide detector was measured to drift by 0.004 dB per five minutes when the test was performed immediately after filling the InSb dewar with liquid nitrogen. The same test was repeated 1.5 hours after filling the dewar, and the drift was reduced to 0.002 dB per five minutes, indicating the need to prefill the dewar approximately 2 hours before running any critical tests.
2.0 How the IR System 1 differs from the Photon Kinetics FOA-2000

The NRL IR System 1 differs from the commercially available Photon Kinetics FOA-2000 with the addition or substitution of several key components. They are:

a. A three-grating monochromator, with a stepper motor to select the grating,

b. custom design diffraction limited infrared lenses,

c. a quartz-tungsten lamp with extended spectral window envelope,

d. additional cutoff filters for the extended range of the monochromator,

e. infrared neutral density filters to accommodate the range of the monochromator,

f. a liquid nitrogen cooled indium antimonide detector for the range from 1.8 μm to 4.0 μm,

g. vacuum chuck V-grooves to accommodate the larger fiber diameter of the infrared fiber,

h. special annular apertures for DMA measurements,

i. a numerical aperture measurement technique that uses a knife edge rather than a pinhole to scan the far field,

j. software that has been adapted to account for the differences in hardware.

2.1. Monochromator

The monochromator is a Jarrell Ash Monospec® 27 with three gratings that may be interchanged by moving a turret. The gratings are summarized as:

<table>
<thead>
<tr>
<th>grating freq.</th>
<th>blaze λ</th>
<th>λ range</th>
<th>avg. dispersion</th>
<th>spectral bandpass</th>
</tr>
</thead>
<tbody>
<tr>
<td>600 gr/mm</td>
<td>1 μm</td>
<td>0.6 to 1.79 μm</td>
<td>6 nm/mm</td>
<td>3 nm</td>
</tr>
<tr>
<td>300 gr/mm</td>
<td>2 μm</td>
<td>1.8 to 2.69 μm</td>
<td>12 nm/mm</td>
<td>6 nm</td>
</tr>
<tr>
<td>150 gr/mm</td>
<td>4 μm</td>
<td>2.7 to 4.0 μm</td>
<td>24 nm/mm</td>
<td>12 nm</td>
</tr>
</tbody>
</table>

The approximate spectral bandpass figures assume that the output slit size is 0.5 mm.

The monochromator wavelength selector is automated with a stepper motor. The step angle on the motor is 1.8 degree per step, and it takes 200 steps for one complete revolution. The Monospec® 27 has an analog wavelength counter which is calibrated to the 1200 groove/mm grating. To obtain the proper wavelength for each grating, the counter reading must be multiplied by 2, 4, and 8 for the 600 groove/mm, 300 groove/mm, and 150 groove/mm...
gratings, respectively. One complete revolution on the wavelength selector corresponds to 25 nm of wavelength scan for a 1200 groove/mm grating. The gear ratio between the monochromator and stepper motor is 3:1. The number of steps on the motor required to scan 1 nm using the 1200 groove/mm grating can be calculated by the formula,

\[ G \cdot \frac{N_m}{W_s}, \]

where \( N \) is the number of motor steps per revolution, \( W \) is the wavelength scan per revolution on selector, and \( G \) is the gear ratio between the monochromator and stepper motor. The 1200 groove/mm grating requires 24 steps for a 1 nm scan. The 600 groove/mm, 300 groove/mm, and 150 groove/mm gratings require 12 steps/nm, 6 steps/nm, and 3 steps/nm, respectively.

2.2. Infrared Lenses

The infrared lenses were designed and constructed by Infrared Optics, Inc. of Farmingdale, NY. They are multi-element lenses fabricated of barium fluoride and lithium fluoride, or zinc sulfide. The lenses were designed to correct for most spherical aberrations between 0.6 and 4.0 \( \mu \)m, with a spot size of 35 \( \mu \)m at 2.5 \( \mu \)m wavelength. The physical dimensions and focal lengths of the lenses were designed to be identical to the standard lenses used in the FOA--2000, in order to facilitate their replacements. The lens parameters are summarized in Figure 2.

2.3 Quartz-Tungsten Lamp

A Ushio Model no. JC12V-50W H2O G/1.0 tungsten halogen lamp is used for the white light source. This lamp utilizes a special quartz envelope which has an extended transmittance out to 4.0 \( \mu \)m.

2.4 Cutoff Filters

Since the Jarrell-Ash monochromator has a greater spectral range than the original FOA--2000 monochromator, it was necessary to add additional cutoff filters to eliminate second- and higher-order spectra from the longer wavelengths. The filter numbers and their cut-on wavelengths are listed below.

<table>
<thead>
<tr>
<th>Filter No.</th>
<th>Cut-on Wavelength</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>540 nm</td>
</tr>
<tr>
<td>2</td>
<td>850 nm</td>
</tr>
<tr>
<td>3</td>
<td>1525 nm</td>
</tr>
<tr>
<td>4</td>
<td>2175 nm</td>
</tr>
<tr>
<td>5</td>
<td>3150 nm</td>
</tr>
</tbody>
</table>
2.5 Infrared Neutral Density Filters

The neutral density filters in the original FOA-2000 are specified only for operation over the limited spectral range of that instrument. They were replaced in the NRL IR System 1 with neutral density filters designed for use in the infrared up to 4.0 μm.

The attenuation of each filter was measured over the range from 800 to 4000 nm using a modified version of the FOA-2000 QC software subroutine SUB Attn_calib. The results of the ND filter calibration tests are given in Appendix A. The attenuation is not very uniform over the spectral range. The ND filters are not used in any spectral attenuation, differential modal attenuation, or numerical aperture tests on the instrument, because of the low radiance of the lamp, eliminating the need to attenuate the output of the lamp. It is possible that the calibration values given there could be incorporated into a look-up table in the system software, such that any time a ND filter is used at some wavelength, then the measured attenuation of that filter at that wavelength is recalled for use in calculations. The original FOA-2000 software however does not easily lend itself to incorporating such a feature, so that including it would entail an effort of moderate difficulty.

2.6 InSb Detector

A liquid nitrogen-cooled indium antimonide (InSb) detector manufactured by Infrared Associates is used to cover the spectral range from 1800 to 4000 nm. The system software automatically switches between the Ge detector and the InSb detector at 1800 nm. The preamplifier used for the InSb detector is an Infrared Associates model PPA-15-IS. The schematic for the InSb detector power supply is given in Appendix B.

2.7 Vacuum Chucks for 200 μm OD fiber

The original FOA-2000 vacuum chucks can only accommodate fibers with outside diameters (OD) up to 140 μm. These chucks have been replaced with ones that can accommodate fibers with up to 200 μm OD.

2.8 Annular Apertures for Differential Modal Attenuation Measurements

Numerical aperture launch restrictors have been adapted to include annular apertures, which are used for launching high order modes in step index fibers, for differential modal attenuation tests. Launch conditions are restricted by manually placing apertures that restrict the launch numerical aperture (NA) to a narrow range, thereby exciting a limited mode group in
2.9 Knife edge numerical aperture measurements

Due to the low radiance of the lamp source, the measurement procedure uses a scanning knife-edge, rather than a scanning aperture in the far-field as specified in EIA FOTP #47. The knife edge technique is an adaptation of an EIA procedure for determining the mode field diameter of a single mode fiber. In this procedure, a knife edge is scanned across the far field output of the fiber, and a lens is used to collect the light passed by the knife edge and direct it to the detector. The computer reads the output of the detector at the lock-in amplifier, which is effectively the integrated power as a function of far field angle. This data is then differentiated and smoothed to yield the far field radiation pattern of the fiber. From this far field the numerical aperture is derived. A criterion of 5% of maximum intensity is used to determine the numerical aperture.

2.10 Major changes in the FOA-2000 software

2.10.1 DMA Measurement

The software has been rewritten so that multiple wavelength scans can be performed in such a way that only one cutback is required. A few new subroutines were created to enable this change in the measurement procedure. A description of their operation is included in the discussion below.

SUB Fibertest2: As with the Far Field test, the user is first queried as to the source of the data which he wishes to view. That is, he may indicate that a new test is to be performed, or that data from a previous run is to be reviewed. Previous data may either be data which was collected earlier in the day (computer on continuously) and is present in the dynamic memory buffer, or data which is stored on a diskette. This query takes place by calling the subroutine FNDatasource, which returns a 0 if a new test is desired, 1 if memory in the buffer is desired, or 2 if the routine Retrieve is to be called to access data on diskette. If either 1 or 2 are returned to Fibertest2, data is loaded into the array called "Dmaattendata", the test portion of the routine is skipped, and the data is plotted on the screen. More details about the plot are below.

SUB Dmarun: First the user is queried about which numerical aperture restrictor to use (including #0 = no restrictor). This is performed by calling the routine FNGetrestrictor, which first lists the restrictor numbers and their corresponding NA range, then uses the FNGetint to determine and return the (integer) restrictor number. The first restrictor number is stored in the (0,1) position of the array "Dmarundata" (see supplemental sheet 1), while subsequent numbers, up to 11, are stored in (0,2), (0,3), and so on.
The wavelengths to scan, and the total number of wavelengths to scan, \( n \), are shared with this routine through the common block /Wavelength/ command. The number of wavelengths is stored in the (0,0) position of "Dmarundata." The (1,0) position of this array contains the fiber length, while the (2,0) position holds the number of DMA runs performed. The wavelength scan is then performed on the long or "run" piece of fiber, and the voltages are stored in the column beneath the restrictor number, in the row corresponding to the wavelength at which the voltage was read.

After each before-the-cutback scan, the user is asked to see if another DMA run (i.e., another NA range) is desired; if yes, the new restrictor number is requested, time given to insert the restrictor, and the program returns to the wavelength scan portion and continues as before.

If no more NA ranges are desired, the user is directed to cutback the fiber, being careful to leave the input end undisturbed. The routine Outalign is then called to align the output end. The first restrictor used before cutback is requested, and a short or "ref" fiber wavelength scan is performed. The voltage data here is now stored in columnar form in an array called "Dmarefdata," which has the same (0,0), (1,0), (2,0) entries as "Dmarundata."

It should be understood that before the first wavelength scan on the long fiber, the signal on the detector (through the fiber) is read using the LED source. This is taken as an alignment reference. After the scan for each DMA run, the alignment is again checked, and if it has varied by more than 1%, the user is informed and given the choice of re-doing that particular scan, exiting the test, or proceeding. The same type procedure is used to insure integrity of the input fiber end alignment after the cutback is performed.

**SUB Dmacomp:** The data from the two arrays are passed through a COM statement to this routine, where the calculation is performed to determine the spectral attenuation for each NA range (represented by respective restrictor number). This outcome of the computation is stored in an array named "Dmaattendata," while the wavelengths used for the scan are stored in the positions (1,0) - (n,0), i.e., the first column of this array. As explained in the supplemental sheet 2, the (0,0) position of this array contains both the number of wavelength scans \( n \), and the number of DMA runs performed, \( m \).

**SUB Dmaplotprep:** Next the user is queried as to which column of data he wishes to view (i.e., which restrictor). His choice column is loaded into the second column of the array Specattdata, while the wavelengths are loaded into the first column. The fiber ID number, including the restrictor number and fiber length are loaded into the string Specatt_id$. Then the routine Specatplot is called to plot the particular column of data requested, and operates in the same way as an ordinary spectral attenuation plot as
described by Photon Kinetics in their software listing remarks. If at any point the "STORE DATA" option is entered, the program exits the plot, enters the Archive subroutine, and stores the contents of the array "Dmaattendata," and then returns the user to the main menu. To further review data, such as the loss results of other restrictors, the "Recall data" option must be chosen from the DMA menu.

2.10.2 Far Field Measurement

This test was changed to run from its own menu, and allow the user the option to smooth the data by averaging a variable number of points. The attached flow chart may aid in understanding the logic.

Presuming a new test is specified, the knife edge will scan across the collimated far field pattern, moving to gradually cut off the power reaching the detector. A normalized version of the data from this scan is held in an array called "Farfield;" it is considered the "raw" data, and is plotted against scanner position. This data can be differentiated to obtain an actual far field output pattern, or can be smoothed directly. When the Differentiate option is chosen, the raw data is first loaded into an array called "Ffrawdata." Then it is differentiated, corrected to account for a small angle approximation, renormalized, and stored in an array "Ffdiffdata," which is again plotted on screen. In addition, the routine Numaper is called to calculate the numerical aperture, which is displayed below the plot. At this point the user may smooth the differentiated data, or return to the raw data plot. If the Smooth Data option is chosen, he is asked for the number of points to average, the data smoothing routine Ffsmooth is called, and the smoothed, normalized version of the data is stored in an array called "Ffsmoothdata." This is plotted on the display, along with a recalculated value of numerical aperture. Further smoothing may then be performed on the raw, differentiated, or smoothed data, and plotted accordingly.

It may be noted that when the raw data is smoothed, at present the smoothed version may not be differentiated, only viewed. Because the "smoothed" raw data actually appeared to be less smooth than the actual raw data, and because of ambiguities introduced into the numerical aperture calculation due to smoothing, this was not further modified, though is would be relatively straightforward to do so. Furthermore, an alternate routine has been sketched out (included) by George McCabe which would fit the data to a Gaussian distribution and look for the 5% points there. In the end, this might offer a more repeatable method to obtain a value for the numerical aperture.

2.10.3 Program Organization

Lines relevant to unused tests were purged in many, but not all, places in the system software. Large blocks such as the FibertestX subroutines and
associated routines were deleted, but remnants exist in other places due to not wanting to alter the "foasetup" file and the way it is stored. All lines relevant to the Near Field test were retained.

2.10.4 Fiber Alignment

The Inalign and Outalign routines have been altered so that the first time either are called, the alignment is performed, and a counter variable is set equal to 1. At the end of the alignment, the final voltage on the Ge detector is read and held in the first position of a variable array, called Sig(1). The routine then returns and aligns the fiber a second time, and holds the new final voltage in Sig(2). These two voltage values are then compared, and if they differ by more than 1%, the user is told so, and given the option of continuing anyway, or returning to re-align. If the latter is chosen, then eventually Sig(3) and Sig(2) are compared as before and so on. At present, the maximum number of alignments that can be performed in this way is 10.
3.0. Normal Operating Procedures

This section describes the daily procedures required for proper normal operation of the system.

3.1. Turn-on procedure

The proper sequence for bringing the system up is outlined below.

a. Turn on the FOA-2000 control panel by turning the key on the front panel.

b. Turn on the EG&G 5207 lock-in amplifier.

c. Turn on the red (illuminated) switch on the power supply.

d. Check the voltage of the preamp batteries in the power supply chassis. To do this, switch the small toggle switch labeled "Meter" to either 1 or 2. There are two sets of batteries, labeled 1 and 2, respectively. One set is normally connected to the preamp while the other is held in reserve, or is being recharged. The voltage of the set in use should read greater than 11 volts. Switch the batteries on by turning the switch labeled "Preamp Batteries" to the set with the higher voltage. If the other set shows a voltage below 11 volts, connect the two battery chargers to the connectors labeled "CHARGERS" on the back of the power supply unit. The reserve batteries will be charged automatically.

Important: Switch the "Meter" switch to "off" after checking the batteries. If it is left on, the discharge of the batteries will be accelerated.

e. Make sure that the InSb detector dewar is filled with liquid nitrogen. When refilling the dewar, it is not necessary to shield the detector from room light. The lens that covers the detector face does not transmit light of a wavelength below 1.0 μm. For minimal drift, the dewar should be filled two hours before any important measurements.

f. Load and run the system software, as describe below.

3.2. Software set-up

Place the disk labeled "DISK #1 (AUTOBOOT)" in the left hand drive (drive 0) of the computer. Turn the monitor, disk drive, printer, and computer on. When the computer prompts, remove the autoboot disk and replace it in drive 0 with the system software disk. The system software will execute automatically, and present the user with a menu.
3.3. Fiber End Preparation

For proper use of the vacuum fiber chucks and the elastomeric fiber clamps, it is necessary to strip at least two inches of jacket from the fiber end. The best location for positioning the clamp on the fiber can be gauged by using the two short strips of black tape on the fiber shelf. The distance from the tape to the edge of the fiber shelf is the proper length of bare fiber that should extend from the fiber clamp.

3.4 Important Commands

Here we list several convenient commands that may be executed from the HP computer.

<table>
<thead>
<tr>
<th>Command</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop</td>
<td>To stop execution of the system software because something is wrong.</td>
</tr>
<tr>
<td>Call Menu</td>
<td>To access the main menu.</td>
</tr>
<tr>
<td>Call Nextwave</td>
<td>To have the unit set to a particular wavelength. For example, to set the wavelength to 850 nm, execute “CALL WAVELENGTH(850)”. See the note “IMPORTANT” below.</td>
</tr>
<tr>
<td>Call Fibertest1</td>
<td>To run the spectral attenuation test directly from the keyboard without having to access it from the main menu.</td>
</tr>
<tr>
<td>Call Fibertest2</td>
<td>To run the differential modal attenuation test directly from the keyboard.</td>
</tr>
<tr>
<td>Call Fibertest3</td>
<td>To run the numerical aperture test directly from the keyboard.</td>
</tr>
<tr>
<td>Call Clearup</td>
<td>Clears the GPIB bus and resets the phase on the lock-in amplifier.</td>
</tr>
</tbody>
</table>
| Run           | To reinitialize the FOA-2000 control panel. This must be done each and every time the control panel is turned off and on again. If the FOA-2000 control panel is turned on, then the system software must be restarted in order to reinitialize the Z-80 microprocessor in the FOA-2000 control panel. In order to do this, stop the program execution (it may be necessary to hit the break
key), and then type "RUN" and press the return key. It is not sufficient to "CALL MENU." The system software must be restarted. After restarting the software, the "Equipment Preset" routine should be run.

Call F2000send ("INSB")

Connects the InSb detector to the lock-in amplifier, and sets the mirrors to direct the fiber output to the InSb detector.

Call Cleardata To clear all data from the memory buffer. This should be called before running the first test on a new fiber if other fibers were run since the machine was turned on.

IMPORTANT: To set the monochromator to a desired wavelength, it is necessary to use the "CALL NEXTWAVE" command instead of using the front panel control. The system software will then insure that the correct grating, cutoff filter, and detector are set up. In addition, the software calculates the proper setting for the monochromator shaft and automatically sets it there. The control panel should only be used to make minor (<50 nm) adjustments in the wavelength displayed on the control panel LED display.
4.0. Maintenance

4.1 Alignment Procedure for NRL IR System 1

4.1.1 Definitions:

The x direction is perpendicular to the beam direction, and parallel to the bench surface. The y direction is perpendicular to both the beam direction and the bench surface. The z direction is parallel to the beam direction.

The IR target is an aluminum piece with cross-hair lines etched on it. Its base should just fit into the milled slots (to assure lateral consistency), and have a cross-hair marked directly over the center of the slot at a height of 1 3/8 inches above the surface of the bench (not the slot).

4.1.2. Main LED Beam Path.

a. Remove lenses 1, 2, 3, 4 from the bench. Select LED on the front panel, with the launch spot out. Using the IR target, align the LED beam all the way around the bench to the camera. Start by engaging LASER 1 on the front panel. When this is done, the mirror in BS2 will switch out to allow the beam to pass through BS2. In addition, the stepper motor driving the monochromator turret will attempt to turn the shaft. This is expected, so don’t be concerned by the sound. Put the IR target in the milled slot at position A, and adjust the LED lens 7 in x and y to align the LED output to the cross-hairs. Adjust the lens in z in order to collimate the beam as well as possible.

b. Next disengage LASER 1 to bring the lower mirror in BS2 back up, directing the beam towards BS4. Again align the beam to the IR target. Now engage THRU TRANSMISSION with the output target out, and adjust BS4 lower mirror to direct the beam onto the IR target at C. Engage FIBER OR SOURCE and proceed to align to the IR target at D. It may be necessary to re-adjust the collimation by tweaking the z position of the LED lens 7. Next adjust mirror M1 to collect as much of the beam as possible, and direct it towards the target at position F. The 275 mm lens 11 should still be in place, roughly half way between M1 and M2. Finally, adjust M2 to direct the beam into the video camera. Leave the image of the LED on the right one third of the monitor, centered vertically.

4.1.3. Input Objective Lens.

a. Center the input objective lens 5 in x, y, and z over the range of travel of the respective motors. To do this, first push the appropriate button on the front panel to engage the corresponding motor. Then turn the front panel knob until the front panel display shows four dots, indicating that the motor has reached the end of its range of travel. Then press the ZERO button to zero the
display. Next, while holding the local button down, rotate the front panel knob in the opposite direction (from the previous motion) until the four dots vanish. Release the local button and continue turning the knob until the four dots appear again, indicating the limit of travel in the opposite direction. Take the reading on the front panel display and divide by two (if the display “tripped over” to 000, be sure to add 1000 to the number before division). Hold the local button down, and rotate the the knob in the opposite direction until the dots vanish. Then move the knob until the display shows the number that resulted from the division by two. Press ZERO to re-zero the display. This position is the center of travel in the appropriate axis. Repeat this procedure for the remaining two axes.

b. Place the lens cap (with white target and mark in center) on the lens. Position the lens in x, y, and z so that the center of the LED beam strikes the center of the lens cap. Note the reading on the front panel in x and y, which displays how many units away from the center of the lens travel the LED beam is.

c. If the reading in x or y is more than about 150 units, the brass U-shaped bracket in the opto-sensor may need to be adjusted to redefine the lens travel limits and therefore the lens travel zero. In order to do this, remove the bracket holder (x axis is underneath bench, y is above), and adjust the position either up or down slightly. Repeat steps a and b until the LED beam corresponds to the center of the x and y lens travel to within acceptable limits.

d. Redefine the zero lens position at the center of the LED beam by pressing the ZERO button on the front control panel for each input lens motor.

4.1.4. Input Fiber Chuck.

a. Remove the lens cap from lens 5. Make sure step 2d has been taken.

b. Prepare a length of fiber (1-2 meters) with cleaved ends. Place one end in the input fiber chuck, and place the other end of the fiber in a power meter (Si detector).

c. Loosen the set screws holding the vacuum chuck and adjust the vacuum chuck to maximize the power injected through the fiber, as detected by the power meter. To adjust horizontally, move the vacuum chuck horizontally. Make an effort to keep the chuck axis perpendicular to the lens. To adjust vertically, use the front panel control (input-y). To adjust longitudinally, push the fiber in and out for coarse adjustments, and use the front panel (input-z) for fine adjustments.

d. Tighten chuck screws so that the chuck is locked firmly in place.
4.1.5. Launch Spot.

a. Engage the BS4 lower mirror by pressing SOURCE. Replace lens 2 on the bench. Set launch spot into the beam by engaging LAUNCH SPOT. Move lens 2 along the slot (in z) to focus the spot on the monitor.

b. Disengage the launch spot. Replace lens 1 (in the adjustable mount) on the bench and move it along the slot to focus the LED on the monitor. Adjust the lens mount in x and y to center the LED image over its previous position on the right one third of the monitor, centered vertically.

c. Engage the launch spot again. Adjust the aperture position on the launch spot carriage in x and y to center the spot over the center of the LED image.

4.1.6. Output Objective Lens.

a. Engage the beamsplitter mirror in the top of BS4 by pressing FIBER LOAD. Loosen the mirror and adjust it until a (probably dim) reflected image of the input fiber end is positioned over the LED image (right one third, centered vertically). This squares the position of the light reflected onto the output objective lens.

b. Follow the procedure of Step 2 to center the output objective lens within its range of travel. Note however that because of the nature of the beamsplitter, in this case the LED beam will appear as a half-circle only. Be sure to redefine the zero lens position for each output lens motor.

c. Re-adjust the upper beamsplitter mirror of BS4 to direct the input fiber image onto the left third of the monitor, centered vertically (over the grease pencil marks). Tighten the mirror screws to lock it into place.

4.1.7. Output Fiber Chuck.

a. Place one end of the prepared fiber in the output chuck. Inject white light into the other end (this may be simply accomplished by placing the fiber end near the filament of an ordinary light bulb).

b. Press THRU TRANSMISSION to allow the white light through the fiber to reach the camera. Loosen the set screws holding the output vacuum chuck and adjust the chuck horizontally to put the output fiber image over the LED image in the right one third of the screen. Center the image vertically by adjusting the front panel control (output-y). To focus the image, push the fiber in and out for coarse adjustments, and use the front panel (output-z) for
c. Place the IR target at the position of lens 3 and check that fiber output is aligned with crosshairs. Then place IR target just after lens 11 and insure that beam is still aligned with crosshairs. If not, move the output fiber chuck angularly in x, and then reiterate steps b and c until the output fiber path lies squarely over the line between BS4 and BS5.

4.1.8. Output Target.

a. Replace lens 4 on the bench. Engage the output target. Move lens 4 along the slot (in z) to focus the image of the output target on the monitor.

b. Disengage the output target. Replace lens 3 (in the adjustable mount) on the bench and move it along the slot to focus the LED on the monitor. Adjust the lens mount in x and y to center the LED image over its previous position on the right one third of the monitor, centered vertically.

c. Engage the output target again. Position the aperture on the output target carriage in x and y over the LED image.

d. At this point the LED, output fiber, and output target should all be focussed on the same position on the right one-third of the screen, centered vertically. This position should be remarked with a grease pencil if necessary.

4.1.9. Lamp Path.

a. Engage LED on the front panel. Switch the lamp on, if it's not on already. Put the IR target at position F in the milled slot that leads from the monochromator output and BS3. Adjust lens 8 to focus the monochromator output on the IR target. Also adjust the positioning knob on top of the lamp to maximize the amount of light into the monochromator, and onto the IR target.

b. Engage the LAMP on the front panel. Engage the button below “LASER THREE” on the front panel. As before, BS2 will switch mirrors and the stepper motor driving the monochromator turret will attempt to turn the shaft. Put the IR target in the milled slot at position A. Adjust the mirror in BS3 to align the monochromator output to the crosshairs as well as possible. Engage LED again.

4.1.10. Ge Detector.

a. Engage SOURCE, Ge DET, and DETECTOR on the front panel control. With
the attenuator at 0, an image of the LED reflected from the surface of the Ge detector should be visible on the monitor. Adjust lens 10 (on the detector module) in x, y, and z to roughly center and focus the image within the large area of the detector.

b. Disengage the DETECTOR switch (upper beamsplitter mirror on BS5), and maximize the output of the Ge detector as seen on the lock-in amplifier.

c. Re-engage DETECTOR and be sure the image of the LED is not too near the edge of the Ge detector. (The most sensitive spot on the detector appears to be near the upper left edge.) Finally, disengage the DETECTOR switch.

4.1.11. InSb Detector.

a. In order to engage the InSb detector, the FOA-2000 System Software must have been loaded onto the HP computer. If the program is running (e.g. a menu is displayed on the HP screen), first press STOP on the keyboard. To connect the detector output to the lock-in amplifier, type the command

\[ \text{CALL F2000send("INSB")} \]

b. Adjust lens 9 on the InSb detector module in x, y, and z to maximize the output of the detector as seen on the lock-in display. Large adjustments in x and y are not recommended.
5.0. How to get help.

In the event that the system does not appear to operate correctly, or if the HP computer returns an error message, the appropriate sequence of actions is as follows:

1. Review the section below entitled "Likely Problems" to see if the fix is indicated there.

2. If the computer indicates an FOA-2000 error message, check page 12-16 of the Photon Kinetics Installation manual for an explanation of the problem.

3. If the computer indicates an HP software problem, then check the "Error Message" appendix of the HP Language Reference manual for an explanation of the error.

If the above steps do not remedy the situation, then contact Russ May or Rick Claus at the Fiber & Electro-Optics Research Center, Virginia Tech, at (703) 231-7203. Replacement parts and knowledgeable insight into the correct operation of the original, unaltered FOA-2000 may be obtained by calling Customer Support at Photon Kinetics, Beaverton, OR, at (503) 644-1960. It should be made clear to Photon Kinetics that the instrument under discussion is Serial No. E4221, which was adapted for use with fluoride fiber by Virginia Tech.

5.1 FOA-2000 Error Messages:

Occasionally the HP computer will indicate an "FOA-2000 error" together with an error number. Most often this may occur when the computer will mistakenly try to drive a stepper motor beyond its permissible range. The meaning of the error number may be found on page 12-16 of the FOA-2000 Installation manual.

Some of the system software routines poll the instrument status of the EG&G lock-in amplifier. If an error is returned by the lock-in to the HP computer, the routine will indicate an error and report the HP basic error number. A description of the error and the associated number is found in the "Error Message" appendix in the back of the HP Basic Language Reference manual.
5.2 Likely Problems

A list of likely problems, their possible causes, and remedies follow:

<table>
<thead>
<tr>
<th>Problem</th>
<th>Possible Cause and Remedy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer displays &quot;Division by zero&quot; error; or output graphs show measurements to be very noisy.</td>
<td>a. Preamp power supply not turned on. b. Lamp power supply not switched on. c. Lamp bulb burned out. d. Preamp battery charge low. e. The phase lock may have been lost on the phase-lock amplifier. This appears to happen after the GPIB bus has been cleared with a &quot;CLEAR 7&quot; command. To reset the phase of the amplifier, type &quot;CALL CLEARUP&quot; and Return. If the program has been halted, then type &quot;CALL MENU&quot; to access the main menu. f. Tried to read archived data in from a non-existent file.</td>
</tr>
</tbody>
</table>

FOA Error no. 97

If the FOA-2000 control panel is turned on, then the system software must be restarted in order to reinitialize the Z-80 \( \mu \)processor in the FOA-2000 control panel. In order to do this, stop the program execution (it may be necessary to hit the break key), and then type "RUN" and press the return key. It is not sufficient to "CALL MENU." The system software must be restarted. After restarting the software, the "Equipment Preset" routine should be run.

The HP computer seems to have halted or is "stuck" while trying to issue a command to the FOA-2000 unit or the lock-in amplifier. press the return key. Then type "CALL return key. This will the lock-in when a It may be possible by typing it will be and restart the

The GPIB bus may have crashed when program execution was halted while the computer was issuing a command or waiting for a status byte on the bus. To remedy, first press the "STOP" key on the computer. Then type "CLEAR 7" and CLEARUP" and the cause the phase to be reset on amplifier, which is often lost "CLEAR 7" is executed. to continue program execution "CONTINUE", but more likely necessary to "CALL MENU" test from the beginning.
Can’t see the fiber end in “FIBER LOAD” mode.

- b. Fiber end off the screen. Put the FOA-2000 in “VIDEO OUT” mode, and peak the reading on the lock-in amp as the fiber is manually adjusted using the fiber movement controls on the FOA-2000 control panel. Then return the FOA-2000 to “FIBER OR SOURCE” mode.

During auto-alignment, the computer consistently returns messages indicating that the fiber end positions need to be adjusted.

- a. The fiber end might not be perpendicular to the fiber axis. Check the end angle using a fiber inspection scope, or recleave the fiber.
- b. The fiber alignment motors may need to be recentered. See page 13-1 of the FOA-2000 installation manual for a procedure to recenter the motors.

Grinding sound from monochromator

In this case, the computer has lost track of actual position of the monochromator shaft, and is attempting to drive the shaft past its limits. The grinding sound results from the stepper motor slipping. To remedy, turn off the key switch on the FOA-2000 control panel immediately. Then turn on the front panel again, and type and execute “RUN” on the HP computer. Then run the “EQUIPMENT PRESET” subroutine from the main menu.
Appendix A. Neutral Density Filter Calibration Results
SPECTRAL ATTENUATION

FIBER ID: Attenuator calibration for ATTN $1$ 17-MAR-90 14:26:41
LENGTH: 0 km

<table>
<thead>
<tr>
<th>WAVELENGTH</th>
<th>ATTENUATION (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>800</td>
<td>-11.56</td>
</tr>
<tr>
<td>850</td>
<td>-11.14</td>
</tr>
<tr>
<td>900</td>
<td>-9.83</td>
</tr>
<tr>
<td>950</td>
<td>-6.91</td>
</tr>
<tr>
<td>1000</td>
<td>-4.89</td>
</tr>
<tr>
<td>1050</td>
<td>-2.55</td>
</tr>
<tr>
<td>1100</td>
<td>-2.69</td>
</tr>
<tr>
<td>1150</td>
<td>-2.11</td>
</tr>
<tr>
<td>1200</td>
<td>-1.76</td>
</tr>
<tr>
<td>1250</td>
<td>-1.65</td>
</tr>
<tr>
<td>1300</td>
<td>-1.45</td>
</tr>
<tr>
<td>1350</td>
<td>-1.49</td>
</tr>
<tr>
<td>1400</td>
<td>-2.07</td>
</tr>
<tr>
<td>1450</td>
<td>-1.87</td>
</tr>
<tr>
<td>1500</td>
<td>-1.96</td>
</tr>
<tr>
<td>1550</td>
<td>-2.46</td>
</tr>
<tr>
<td>1600</td>
<td>-2.41</td>
</tr>
<tr>
<td>1650</td>
<td>-3.83</td>
</tr>
<tr>
<td>1700</td>
<td>-3.39</td>
</tr>
<tr>
<td>1750</td>
<td>-2.21</td>
</tr>
<tr>
<td>1800</td>
<td>-2.21</td>
</tr>
<tr>
<td>1850</td>
<td>-2.17</td>
</tr>
<tr>
<td>1900</td>
<td>-2.11</td>
</tr>
<tr>
<td>1950</td>
<td>-2.03</td>
</tr>
<tr>
<td>2000</td>
<td>-2.98</td>
</tr>
<tr>
<td>2050</td>
<td>-2.92</td>
</tr>
<tr>
<td>2100</td>
<td>-2.87</td>
</tr>
<tr>
<td>2150</td>
<td>-2.82</td>
</tr>
<tr>
<td>2200</td>
<td>-2.75</td>
</tr>
<tr>
<td>2250</td>
<td>-2.71</td>
</tr>
<tr>
<td>2300</td>
<td>-2.66</td>
</tr>
<tr>
<td>2350</td>
<td>-2.60</td>
</tr>
<tr>
<td>2400</td>
<td>-2.54</td>
</tr>
<tr>
<td>2450</td>
<td>-2.48</td>
</tr>
<tr>
<td>2500</td>
<td>-2.42</td>
</tr>
<tr>
<td>2550</td>
<td>-2.34</td>
</tr>
<tr>
<td>2600</td>
<td>-2.32</td>
</tr>
<tr>
<td>2650</td>
<td>-2.24</td>
</tr>
<tr>
<td>2700</td>
<td>-2.23</td>
</tr>
<tr>
<td>2750</td>
<td>-2.12</td>
</tr>
<tr>
<td>2800</td>
<td>-2.11</td>
</tr>
<tr>
<td>2850</td>
<td>-2.11</td>
</tr>
<tr>
<td>2900</td>
<td>-2.11</td>
</tr>
<tr>
<td>2950</td>
<td>-2.09</td>
</tr>
<tr>
<td>3000</td>
<td>-2.07</td>
</tr>
</tbody>
</table>
SPECTRAL ATTENUATION

FIBER ID: Attenuator calibration for ATTN #2 17-MAR-90 14:34:59
LENGTH: 0 km

WAVELENGTH   ATTENUATION (dB)

800      -22.56
850      -21.82
900      -20.25
950      -19.98
1000     -14.60
1050     -12.92
1100     -11.73
1150     -10.88
1200     -10.30
1250     -9.98
1300     -9.59
1350     -9.50
1400     -9.98
1450     -9.62
1500     -9.62
1550     -14.47
1600     -14.80
1650     -15.71
1700     -15.50
1750     -15.44
1800     -15.37
1850     -15.28
1900     -15.20
1950     -15.10
2000     -15.04
2050     -14.99
2100     -14.94
2150     -14.89
2200     -14.84
2250     -14.81
2300     -14.78
2350     -14.77
2400     -14.66
2450     -14.65
2500     -14.64
2550     -14.59
<table>
<thead>
<tr>
<th>WAVELENGTH</th>
<th>ATTENUATION (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>800</td>
<td>-34.31</td>
</tr>
<tr>
<td>850</td>
<td>-33.71</td>
</tr>
<tr>
<td>900</td>
<td>-32.25</td>
</tr>
<tr>
<td>950</td>
<td>-29.15</td>
</tr>
<tr>
<td>1000</td>
<td>-26.99</td>
</tr>
<tr>
<td>1050</td>
<td>-25.42</td>
</tr>
<tr>
<td>1100</td>
<td>-24.26</td>
</tr>
<tr>
<td>1150</td>
<td>-23.45</td>
</tr>
<tr>
<td>1200</td>
<td>-22.87</td>
</tr>
<tr>
<td>1250</td>
<td>-22.54</td>
</tr>
<tr>
<td>1300</td>
<td>-22.15</td>
</tr>
<tr>
<td>1350</td>
<td>-22.02</td>
</tr>
<tr>
<td>1400</td>
<td>-27.42</td>
</tr>
<tr>
<td>1450</td>
<td>-22.06</td>
</tr>
<tr>
<td>1500</td>
<td>-22.00</td>
</tr>
<tr>
<td>1550</td>
<td>-26.65</td>
</tr>
<tr>
<td>1600</td>
<td>-26.85</td>
</tr>
<tr>
<td>1650</td>
<td>-27.68</td>
</tr>
<tr>
<td>1700</td>
<td>-27.40</td>
</tr>
<tr>
<td>1750</td>
<td>-27.20</td>
</tr>
<tr>
<td>1800</td>
<td>-26.80</td>
</tr>
<tr>
<td>1850</td>
<td>-26.44</td>
</tr>
<tr>
<td>1900</td>
<td>-26.30</td>
</tr>
<tr>
<td>1950</td>
<td>-28.20</td>
</tr>
<tr>
<td>2000</td>
<td>-25.08</td>
</tr>
<tr>
<td>2050</td>
<td>-25.01</td>
</tr>
<tr>
<td>2100</td>
<td>-25.89</td>
</tr>
</tbody>
</table>
### SPECTRAL ATTENUATION

**FIBER ID:** Attenuator calibration for ATTN 4 17-MAR-90 14:54:53  
**LENGTH:** 0 km

<table>
<thead>
<tr>
<th>WAVELENGTH</th>
<th>ATTENUATION (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>800</td>
<td>-44.20</td>
</tr>
<tr>
<td>850</td>
<td>-42.82</td>
</tr>
<tr>
<td>900</td>
<td>-40.85</td>
</tr>
<tr>
<td>950</td>
<td>-37.31</td>
</tr>
<tr>
<td>1000</td>
<td>-34.65</td>
</tr>
<tr>
<td>1050</td>
<td>-32.77</td>
</tr>
<tr>
<td>1100</td>
<td>-31.25</td>
</tr>
<tr>
<td>1150</td>
<td>-30.09</td>
</tr>
<tr>
<td>1200</td>
<td>-29.17</td>
</tr>
<tr>
<td>1250</td>
<td>-28.54</td>
</tr>
<tr>
<td>1300</td>
<td>-27.85</td>
</tr>
<tr>
<td>1350</td>
<td>-27.47</td>
</tr>
<tr>
<td>1400</td>
<td>-27.62</td>
</tr>
<tr>
<td>1450</td>
<td>-28.99</td>
</tr>
<tr>
<td>1500</td>
<td>-29.97</td>
</tr>
<tr>
<td>1550</td>
<td>-31.06</td>
</tr>
<tr>
<td>1600</td>
<td>-31.33</td>
</tr>
<tr>
<td>1650</td>
<td>-31.71</td>
</tr>
<tr>
<td>Year</td>
<td>Value</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
</tr>
<tr>
<td>1800</td>
<td>-29.61</td>
</tr>
<tr>
<td>1850</td>
<td>-29.09</td>
</tr>
<tr>
<td>1900</td>
<td>-29.82</td>
</tr>
<tr>
<td>1950</td>
<td>-29.62</td>
</tr>
<tr>
<td>2000</td>
<td>-29.35</td>
</tr>
<tr>
<td>2050</td>
<td>-29.19</td>
</tr>
<tr>
<td>2100</td>
<td>-28.90</td>
</tr>
<tr>
<td>2150</td>
<td>-28.76</td>
</tr>
<tr>
<td>2200</td>
<td>-28.60</td>
</tr>
<tr>
<td>2250</td>
<td>-28.48</td>
</tr>
<tr>
<td>2300</td>
<td>-28.29</td>
</tr>
<tr>
<td>2350</td>
<td>-28.06</td>
</tr>
<tr>
<td>2400</td>
<td>-27.81</td>
</tr>
<tr>
<td>2450</td>
<td>-27.57</td>
</tr>
<tr>
<td>2500</td>
<td>-27.30</td>
</tr>
<tr>
<td>2550</td>
<td>-27.16</td>
</tr>
<tr>
<td>2600</td>
<td>-27.24</td>
</tr>
<tr>
<td>2650</td>
<td>-27.00</td>
</tr>
<tr>
<td>2700</td>
<td>-26.86</td>
</tr>
<tr>
<td>2750</td>
<td>-26.62</td>
</tr>
<tr>
<td>2800</td>
<td>-26.39</td>
</tr>
<tr>
<td>2850</td>
<td>-26.15</td>
</tr>
<tr>
<td>2900</td>
<td>-26.00</td>
</tr>
<tr>
<td>2950</td>
<td>-25.65</td>
</tr>
<tr>
<td>3000</td>
<td>-25.40</td>
</tr>
<tr>
<td>3050</td>
<td>-25.15</td>
</tr>
<tr>
<td>3100</td>
<td>-24.90</td>
</tr>
<tr>
<td>3150</td>
<td>-24.65</td>
</tr>
<tr>
<td>3200</td>
<td>-24.40</td>
</tr>
<tr>
<td>3250</td>
<td>-24.15</td>
</tr>
<tr>
<td>3300</td>
<td>-23.90</td>
</tr>
<tr>
<td>3350</td>
<td>-23.65</td>
</tr>
<tr>
<td>3400</td>
<td>-23.40</td>
</tr>
<tr>
<td>3450</td>
<td>-23.15</td>
</tr>
<tr>
<td>3500</td>
<td>-22.90</td>
</tr>
<tr>
<td>3550</td>
<td>-22.65</td>
</tr>
<tr>
<td>3600</td>
<td>-22.40</td>
</tr>
<tr>
<td>3650</td>
<td>-22.15</td>
</tr>
<tr>
<td>3700</td>
<td>-21.90</td>
</tr>
<tr>
<td>3750</td>
<td>-21.65</td>
</tr>
<tr>
<td>3800</td>
<td>-21.40</td>
</tr>
<tr>
<td>3850</td>
<td>-21.15</td>
</tr>
<tr>
<td>3900</td>
<td>-20.90</td>
</tr>
<tr>
<td>3950</td>
<td>-20.65</td>
</tr>
<tr>
<td>4000</td>
<td>-20.40</td>
</tr>
</tbody>
</table>
Appendix B. InSb Detector Preamp Power Supply Circuit

[Diagram of the circuit with labels for Preamp Battery Switch, Preamp +12 V, + charger A, - charger A, preamp ground, - charger B, preamp -12 V, - charger B, + charger A, preamp +12 V, preamp ground, - charger B, - charger B, preamp -12 V, and Meter Switch]
Appendix C. Index of Technical Reports and Publications

No technical reports other than this final report were generated during the administration of this contract.

There were no publications generated during the administration of this contract.
Appendix D. System Software Listing
REM + PURPOSE:
60 REM + This is the main program that calls all other
62 REM + test and utility modules. It sets up the required data
64 REM + and calls the initialization routines that prepare the
66 REM + FOA-2000 system for measurements. Then, it prints a menu
68 REM + of options for the user to choose from. At present, the menu
70 REM + contains options for running the test sequence defined by
72 REM + the user's FIBERTEST module, setting the time and date,
74 REM + inspecting the system set-up data, and archiving measurement
76 REM + results. Other options can be added easily.
78 REM +
80 REM + First the common data areas are set-up. The data in these common
82 REM + areas are shared among many routines in the utility software. They
84 REM + should not be changed since many routines reference this data.
86 REM +
88 OPTION BASE 0
90 COM /Diskdrive/ Sysdrive$[20],Arcdrive$[20]
92 COM /Ipaths/ @Foa2000,@Egg5205,Tek7854,8Bncdelay,Printer_add
94 COM /Previous/ Previous$[80]
96 COM /Egg5205/ Scales(20),Settle,INTEGRER Num_aver,Range
98 COM /Sysdata/ Serial_num$[40],Lasers(2),Filter_flag,Filters(11),Num_focus,
100 Focus(64,3),Cutoff,Low_wave,High_wave,Det_switch
102 COM /Syscal/ Pin_x,Pin_y,Pin_z,Inx_step,Iny_step,Outx_step,Outy_step,Farfi
104 old step,linoise
After these assignments, all references to the device are made through the appropriate I/O path name. These path names and their characteristics should not be changed unless a device address is changed.

The scale factors for the EGG5207 are stored in a common area called /Eg5207/. These scale factor values are used to scale readings from the EGG5205 into volts. The array is initialized here.

The disk drives are assigned device specifier names used throughout this software when the disk drives are accessed. These drive names are automatically derived from the current MSI. This may be inappropriate for some systems where MSIs are changed to MEMORY or BUBBLE. If this is the case then change these lines to assign these directly. Some examples are:

```
Sys$=SYSTEM$("MSI")
```

```
9836: Sysdrive$=":INTERNAL,4,0" Arcdrive$=":INTERNAL,4,1"
9817: Sysdrive$=":HP9122,701,0" Arcdrive$=":HP9122,700,1"
9816: Sysdrive$=":HP8290X,701,0" Arcdrive$=":HP8290X,701,1"
```
MASS STORAGE IS Sysdrive$.

If the keyboard is a 4620A keyboard then some initialization should be done on it. This is accomplished automatically below.

STATUS KBD,9;Key_id

If BIT(Key_id,5) THEN

CONTROL KBD,14;0 !Set fl=1, not f0

OUTPUT KBD USING "-K",CHR$(255)&CHR$(123) !Put function keys in USER

END IF

If BASIC 4.0 or greater is running, then turn on the display compatibility card if it is there.

IF VAL(SYSTEM$("VERSION,BASIC"))=4.0 AND POS(SYSTEM$("CRT ID"),"B") THEN

CONTROL CRT,21;1

END IF

Now the initialization routine is called to initialize the system.

The system data file is read in, and previously-measured reference data is transferred from the system disk into Common.

Initialization ERROR GOTO Syserror

CALL Sysinit

OFF ERROR

CALL Menu

GOTO Init

We get an error when attempting to call SYSINIT if MAINPROG is run without the rest of the system software package. The loadsub module will build the complete FOA2000 file.

THIS IS FOR PHOTON KINETICS USE ONLY !!

Syserror: IF ERRN=7 THEN

OFF KEY

BEEP

DISP "Do you want to build an F2000SYSTEM file?"

ON KEY 5 LABEL "YES" GOTO Build

ON KEY 6 LABEL "NO" GOTO Done

Wait_here: GOTO Wait_here

Build: OFF KEY

DISP ""

GOTO Init

ELSE

Main_err: BEEP

DISP "MAINPROG -- "&ERRM$

Dead: GOTO Dead1

END IF

Done: DISP ""

END

ELSE

SUB Sysinit

SYSTEM INITIALIZATION MODULE

VERSION 2.1P

COM /Diskdrive/, Sysdrive$, Arcdrive$

COM /Sysdata/, Serial_num$, Laser(*), Filter_flag, Filter(*), Num_focus, Focus (*), cutoff, Low_wave, High_wave, Det_switch


COM /Directref/, Specrefcor(*), Pulserfcor(*), Pulserrefcorwav(*), Correct_fln
COM /Pulserundata/ Pulserundata(*),Pulserun_id$(*)
Pulserunwave(*),Num_a
v$[0],Sys_delay
COM /Jittercor/ Jittercor(*)
COM /Additions/ Curr_wave,Gratings(*),Cur_grating,Wave_step
DIM Filename$[40],Temp(1,256)
ABORT 7 !Send IFC (Interface Clear) on GPIB
GCLEAR
STATUS KBD,9iKey_id
!
IF the number of pulse averages has not been set, default=400
IF Num_avess="" THEN Num_avess="4 0 0"
CALL Init_foa_cntrl !Set the FOA-2000's controller to new mono stuff
!
Now check to see if the user wants to load/reload user routines.
!
ON ERROR GOTO File_error
Load_set:OFF KEY
!
Filename$="foasetup" Set-up file name
DTEP "
INTEGER Index
!
Now read the FOA-2000 set-up file called "foasetup"
!
Readsetup:ASSIGN @Setupfile TO Filename$&Sysdrive$
ENTER @SetupfilelVersion$ !Check the setup file
IF NOT POS(Version$,"VERSION") THEN
BEEP
PRINT TABXY(5,10)"The set up file on this disk is the wrong version.";
PRINT TABXY(5,11)"sub SYSTEMDATA should be called to update it, [wave
length limits and"
PRINT TABXY(5,12)"machine serial number I.D. should be added]."
PRINT TABXY(5,14)"Other data will be read anyway."
WAIT 5
OUTPUT KBD USING "$K";"K"
ASSIGN @Setupfile TO *
ASSIGN @Setupfile TO Filename$&Sysdrive$
GOTO Vfile
!
ELSE
Version_num=VAL(Version$(POS(Version$,"VERSION")+9))
!
IF Version_num<2 THEN
PRINT TABXY(5,10)"This set up file is not current."
WAIT 3
OUTPUT KBD USING "$K";"K"
END IF
!
ENTER @SetupfilelSerial_num$ !Get the machine's S/N
ENTER @SetupfileLow_wave,High_wave !Wavelength limits
Vfile:ENTER @SetupfileLaser(*) !Get the laser wavelengths
ENTER @SetupfileFilter_flag !Get the filter/mono. flag
ENTER @SetupfileFilter(*) !Read entire filter table
ENTER @SetupfileNum_focus !Get number of focus values
ENTER @SetupfileFocus(*) !Get the focus table
ENTER @SetupfileOuflof !Cutoff switch wavelength
ENTER @SetupfilePin_x,Pin_y,Pin_z !Get pinhole position
ENTER @SetupfileInx_step,Iny_step !Get Input stage step size
ENTER @SetupfileOutx_step,Outy_step !Output stage step size
ENTER @SetupfileFarfield_step !Get farfield step size
ENTER @SetupfileLnoise !Low freq. detector noise
IF Version_num>2 THEN
ENTER @SetupfileDet_switch !Detector switch wavelength.
ELSE
Dat switch has not set
END IF
Next, read the insertion delay.

Read_delay:Filename$="pulsedelay"
ASSIGN @Delayfile TO "pulsedelay"
ENTER @Delayfile:Sys_delay
ASSIGN @Delayfile TO *

Read_jitter:Filename$="jittercor"
ASSIGN @Jittercor TO "jittercor"
ENTER @Jittercor:Jittercor(*)
ASSIGN @Jittercor TO *

Next read the variable aperture calibration data.

Readvarapcal:Filename$="varapcal"
ASSIGN @Varapfile TO Filename$&Sysdrive$*
ENTER @Varapfile:Varap_sn$
ENTER @Varapfile:Apca_data(*)
ASSIGN @Varapfile TO *

Next, read the Spectral Attenuation direct-spot correction data

Readspeccor:Filename$="speccor"
ASSIGN @Specreffile TO Filename$&Sysdrive$
ENTER @Specreffile:Specrefcor(*)
ASSIGN @Specreffile TO *

Read the Swept frequency direct-spot correction data for all 3 lasers

Readpulse:FOR Jindex=0 TO 2
Filename$="pulsecor"&VAL$(Jindex+1)
ASSIGN @Pulserreffile TO Filename$&Sysdrive$
ENTER @Pulserreffile:Temp(*)
ENTER @Pulserreffile:Pulsecorwave(Jindex)
FOR Index=0 TO 256
Pulsereffcor(0,Index,Jindex)=Temp(0,Index)
Pulsereffcor(1,Index,Jindex)=Temp(1,Index)
NEXT Index
ASSIGN @Pulserreffile TO *

Readpulse_1: NEXT Jindex
GOTO Done

File_error: SELECT ERRN
CASE 56
SELECT Filename$
CASE "foesetup"
BEEP
PRINT TABXY(1,17):"The FOA-2000 set-up file does not exist on the disk in the primary"
PRINT TABXY(1,18):"disk drive. Please insert the system software disk and press PROCEED."
ON KEY 5 LABEL "PROCEED": GOTO Ready

Wait_2: GOTO Wait_2

Ready: OFF KEY
GOSUB Clr_screen
GOTO Readsetup
CASE "pulsedelay"
Sys_delay=40
GOTO Read_jitter
CASE "jittercor"
GOTO Readvarapcal
CASE "varapcal"
GOTO Readspeccor
CASE "speccor"
BEEP
DISP "The ""USER"" file was not found on this disk."
ON KEY 5 LABEL "RE-TRY" GOTO Load_user
IF BIT(Key_id,5) THEN
ON KEY 6 LABEL "LOAD ALTERNATE" GOTO Try_again
ELSE
ON KEY 6 LABEL "LOAD ALTERNATE" GOTO Try_again
END IF
GOTO Wait_3
CASE ELSE
BEEP
DISP Filename$" was not found."
ON KEY 5 LABEL "RE-TRY" GOTO Try Again
IF BIT(Key_id,5) THEN
ON KEY 6 LABEL "LOAD DEFAULT" GOTO Default
ELSE
ON KEY 6 LABEL "LOAD DEFAULT" GOTO Default
END IF
Wait_3: GOTO Wait_3
Try_again:CAT Sysdrive$% Load_alt
Default:Filename$="USER"
GOTO Load User
END SELECT
CASE 7 1 Call to an undefined subprogram
GOTO Skip_del
CASE 80 1 Disk not installed
BEEP
DISP "There is no disk in the disk drive. Please install the disk and press proceed."
ON KEY 5 LABEL "PROCEED" GOTO Proceed
Wait_4: GOTO Wait_4
Proceed:SELECT Filename$
CASE "foasetup"
GOTO Readsetup
CASE "varapcal"
GOTO Readvarapcal
CASE "speccor"
GOTO Readspeccor
CASE "pulsecor1","pulsecor2","pulsecor3"
GOTO Readpulse
CASE "USER"&Sysdrive$
GOTO Load_user
CASE ELSE
GOTO Load_alt
END SELECT
CASE ELSE
BEEP
DISP "SYSINIT -- Error number "&VAL$(ERRN)
Dead1:GOTO Dead1
END SELECT
Clr_screen"OUTPUT KBD USING ";K";K"
RETURN
Done:SUBEND

SUB Systemdata
field_step,Lnoise
field_step,Lnoise

738  DIM Filename$(25),Title$(200),Keys$(1:20):16
740  !Set key label data for,16's and 17's
742  Data16:DATA QUIT,PRINT DATA,MODIFY DATA,SHOW CAL DATA,RETURN TO MENU,PRINT CAL DATA,CHANGE LASERS,USE MONOCHRM.,CHANGE FILTERS,USE FILTERS
744  DATA USE CUTOFF FIL.MOD/DEL CUTOFF,MOD WAVELimit,ADD CORR VALUE,DEL CORR VALUE,MOD CORR VALUE,DISPLAY DATA,FILTER WHEEL,MONOCHROMATER,CHANGE SERIAL #
746  Data17:DATA QUIT,PRINT DATA,MODIFY DATA,SHOW CAL DATA,RETURN TO MENU,PRINT CAL DATA,CHANGE LASERS,USE MONOCHRM.,CHANGE FILTERS,USE FILTERS
748  DATA USE CUT-OFF FIL.,MOD/DEL CUTOFF,MOD WAVELimits,ADD CORR VALUE,DEL CORR VALUE,MOD CORR VALUE,DISPLAY DATA,FILTER WHEEL,MONOCHROMATER,CHANGE SERIAL #

750  !
752  !If computer is a 9816/36 then read the first set of key labels otherwise
754  !read the second set of key labels.
756  !
758  RESTORE Data16
760  STATUS KBD,9!Key_id
762  IF BIT(Key_id,5) THEN RESTORE Data17
764  READ Keys$(*)
766  !
768  !Write_flag is set if any system data is modified, to indicate
770  !that the foasetup file must be purged and re-written.
772  !
774  Write_flag=0
776  !Clear re-write foasetup flag
778  Filename$="foasetup"&Sysdrive$  !Set-up file name
780  INTEGER IndeA
782  !Display the system data on the CRT (excluding calibration data)
784  !
786  GOSUB Sys_display
788  !
790  !Now ask the user what to do
792  !
794  Sys_menu:BEEP
796  ON KEY 1 LABEL Keys$(20) GOTO Change_sn
798  ON KEY 5 LABEL Keys$(1) GOTO Sys_done
800  ON KEY 6 LABEL Keys$(2) GOTO Print_scrn
802  ON KEY 7 LABEL Keys$(3) GOTO Call_mod
804  ON KEY 8 LABEL Keys$(4) GOTO Call_cal
806  Wait_menu:GOTO Wait_menu
808  Change_sn:OFF KEY
810  DISP ""
812  Write_flag=1
814  INPUT "Enter a serial number or new identifier: ",Serial_num$  
816  GOSUB Sys_display
818  GOTO Sys_menu
820  Print_scrn:OFF KEY
822  DUMP ALPHA
824  GOSUB Clr_screen
826  GOSUB Sys_display
828  GOTO Sys_menu
830  Call_mod:OFF KEY
832  GOSUB Sys_modify
834  GOSUB Clr_screen
836  GOSUB Sys_display
838  GOTO Sys_menu
840  Call_cal:OFF KEY
842  GOSUB Sys_cal
844  GOSUB Clr_screen
846  GOSUB Sys_display
848  GOTO Sys_menu
850  Clr_screen:OUTPUT KBD USING ":,K:K" RETURN
880 Sys_display: GOSUB Clr_screen
882 DISP " Clear the display line
884 PRINT TABXY(5,1);CHR$(129);" FOA-2000 SYSTEM CONFIGURATION: MACHINE SERIAL NUMBER ";Serial_num;CHR$(128)
886 PRINT
888 PRINT CHR$(132);"Wavelength Range:";CHR$(128)&"";VAL$(Low_wave);" nm to ";VAL$(High_wave);" nm. Detector switch at ";VAL$(Det_switch);".
870 IF Filter_flag=1 THEN
872 PRINT CHR$(132);"FILTER WHEEL WAVELENGTHS"
874 PRINT CHR$(132);"Filter";CHR$(128);"Wavelength";CHR$(128);"Filter";CHR$(128);"Wavelength";CHR$(128)
876 PRINT "Wavelength";CHR$(128);"Filter";CHR$(128);"Wavelength";CHR$(128)
878 FOR Index=0 TO 3
880 PRINT USING "2X,20,6X,4D,13X,2D,6X,4D,13X,2D,6X,4D":Index,Filter(Index),Index+4,Filter(Index+4),Index+8,Filter(Index+8)
882 NEXT Index
884 ELSE
886 PRINT
888 IF Filter_flag=2 THEN
890 PRINT "Monochromator installed and cutoff filters are being used."
892 ELSE
894 PRINT "Monochromator installed but cutoff filters not being used."
896 END IF
898 PRINT
900 END IF
902 PRINT CHR$(132);"CORRECTION VALUES"
904 PRINT CHR$(132);"Wavelength X Y Z";CHR$(128);"Wavelength X Y Z";CHR$(128);"Wavelength X Y Z";CHR$(128)
906 PRINT "Wavelength X Y Z";CHR$(128)
908 FOR Index=0 TO 21
910 PRINT USING "3X,4D,3X,3O,2X,3O,2X,4D":Focus(Index,0),Focus(Index,1),Focus(Index,2),Focus[Index,3]
912 PRINT USING "4X,4D,3X,3O,2X,3O,2X,4D":Focus(Index+22,0),Focus(Index+22,2),Focus(Index+22,3)
914 IF Index<21 THEN
916 PRINT USING "4X,4D,3X,3O,2X,3O,2X,4D":Focus(Index+44,0),Focus(Index+44,2),Focus(Index+44,3)
918 END IF
920 NEXT Index
922 RETURN
924 END
926 ! Display Calibration Data
928 !
930 Sys_cal:GOSUB Clr_screen
932 DISPLAY"
934 PRINT TABXY(25,1);CHR$(129);" FOA-2000 CALIBRATION CONSTANTS ";CHR$(129)
936 PRINT
938 PRINT TABXY(33,3);CHR$(132);"PINHOLE POSITION";CHR$(129)
940 PRINT TABXY(23,9);"Pin_x:";Pin_x;"Pin_y:";Pin_y;"Pin_z:";Pin_z
942 PRINT TABXY(30,7);CHR$(132);"FIBER STAGE STEP SIZE";CHR$(128)
944 PRINT TABXY(8,9);"Imx_step:";Imx_step;"Imy_step:";Imy_step;"Outx_step:";Outx_step;"Outy_step:"Outy_step
946 PRINT TABXY(25,11);CHR$(132);"FAR-FIELD RESTRICTOR STEP SIZE";CHR$(128)
948 PRINT TABXY(28,13);"Farfield_step:";Farfield_step
950 PRINT TABXY(26,15);"LOW-FREQUENCY DETECTOR NOISE";CHR$(128)
952 PRINT TABXY(28,17);"Lfnoise:";Lfnoise
954 ON KEY 5 LABEL Keys*(G) GOTO Cal_done
955 ON KEY 6 LABEL Keys*(G) GOTO Cal_print
958 Wait_cal=GOTO Wait_cal
Modify the System data

Modify menu

ON KEY 5 LABEL Keys$(7) GOTO Mod_lasers
SELECT Filter_flag
CASE 1
  ON KEY 6 LABEL Keys$(8) GOTO Use mono
  ON KEY 7 LABEL Keys$(9) GOTO Mod_filter
CASE 0
  ON KEY 6 LABEL Keys$(10) GOTO Use_filter
  ON KEY 7 LABEL Keys$(11) GOTO Use_cutoff
CASE 2
  ON KEY 6 LABEL Keys$(10) GOTO Use_filter
  ON KEY 7 LABEL Keys$(12) GOTO Mod_cutoff
END SELECT
ON KEY 1 LABEL Keys$(14) GOTO Add_corr
ON KEY 2 LABEL Keys$(15) GOTO Del_corr
ON KEY 3 LABEL Keys$(13) GOTO Mod_highlow
ON KEY 4 LABEL Keys$(16) GOTO Mod_corr
ON KEY 8 LABEL Keys$(17) GOTO Mod_done

Wait_mod:GOTO Wait_mod
| Modify laser wavelengths
| END

Mod_lasers:OFF KEY
DISP ""
Write_flag=1
BEEP
ON KEY 5 LABEL "LASER 1" GOTO Laser_1
ON KEY 6 LABEL "LASER 2" GOTO Laser_2
ON KEY 7 LABEL "LASER 3" GOTO Laser_3
Wait_las:GOTO Wait_las
Laser_1:OFF KEY
BEEP
INPUT "Enter the new wavelength for laser 1 in nanometers: ",Laser(0)
GOTO Check
Laser_2:OFF KEY
BEEP
INPUT "Enter the new wavelength for laser 2 in nanometers: ",Laser(1)
GOTO Check
Laser_3:OFF KEY
BEEP
INPUT "Enter the new wavelength for laser 3 in nanometers: ",Laser(2)
GOTO Check
Check:DISP "Do you want to change another laser wavelength?"
BEEP
ON KEY 5 LABEL "YES" GOTO Mod_lasers
ON KEY 6 LABEL "NO" GOTO Mod_done
Wait_chk:GOTO Wait_chk
| Modify the filter flag: If set, clear it; if clear, set it.
| END

Use_mono:OFF KEY
DISP ""
Write_flag=1
Filter_flag=0
GOTO Mod_lasers

Modify the filter flag: If set, clear it; if clear, set it.
Modify filter wavelengths

Mod_filter:OFF KEY
1172 DISP " "
1174 BEEP
1176 Write_flag=1
1178 INPUT "Enter the filter number (0-11) you want to change: ",Filter_num
1180 IF Filter_num>11 OR Filter_num<0 THEN GOTO Mod_filter
1182 BEEP
1184 INPUT "Enter the new wavelength: ",Filter(Filter_num)
1186 DISP "Do you want to change more filter wavelengths?"
1188 BEEP
1190 ON KEY 5 LABEL "YES" GOTO Mod_filter
1192 ON KEY 6 LABEL "NO" GOTO Mod_done
1194 Wait_fil:GOTO Wait_fil
1196 |
1198 | Modify a correction value
1200 |
1202 Mod_corr:OFF KEY
1204 DISP " "
1206 Write_flag=1
1208 Try_again:BEep
1210 INPUT "Enter the correction wavelength you want to modify: ",Wavelen
1212 DISP " "
1214 FOR Index=0 TO Num_focus
1216 IF Wavelen=Focus(Index,0) THEN GOTO Get_new
1218 NEXT Index
1226  GOTO Sys_modify
1228  Get_new: BEEP
1230  INPUT "Enter the new X-axis correction value: ", Focus(Index, 1)
1232  BEEP
1234  INPUT "Enter the new Y-axis correction value: ", Focus(Index, 2)
1236  BEEP
1238  INPUT "Enter the new Z-axis correction value: ", Focus(Index, 3)
1240  Focus(Index, 0) = Wavelen
1242  GOTO Mod_done
1244  !
1246  ! Delete a correction value
1248  !
1250  Del_corr: OFF KEY
1252  BEEP
1254  Write_flag = 1 !Set flag to rewrite foasetup
1256  INPUT "Enter the wavelength of the correction value you want to delete", Wavelen
1258  FOR Index = 0 TO Num_focus  !Find the one to delete
1260  IF Wavelen = Focus(Index, 0) THEN GOTO Delete
1262  NEXT Index
1264  BEEP
1266  DISP "SYSTEM DATA -- The specified wavelength is not in the correction table."
1268  OFF ERROR
1270  GOTO Sys_modify
1272  !
1274  ! Delete the correction entry by moving the rest down by 1
1276  !
1278  Delete: FOR Index = Index TO Num_focus - 1
1280  Focus(Index, 0) = Focus(Index + 1, 0)
1282  Focus(Index, 1) = Focus(Index + 1, 1)
1284  Focus(Index, 2) = Focus(Index + 1, 2)
1286  Focus(Index, 3) = Focus(Index + 1, 3)
1288  NEXT Index
1290  Focus(Index, 0) = 0
1292  Focus(Index, 1) = 0
1294  Focus(Index, 2) = 0
1296  Focus(Index, 3) = 0
1298  Num_focus = Num_focus - 1 !And decrement the count
1300  GOTO Mod_done
1302  !
1304  ! ADD A NEW CORRECTION VALUE
1306  !
1308  Add_corr: OFF KEY
1310  DISP ""
1312  IF Num_focus = 19 THEN
1314  BEEP
1316  DISP "SYSTEM DATA -- The correction table is full, delete an entry first."
1318  OFF ERROR
1320  GOTO Sys_modify
1322  END IF
1324  BEEP
1326  Write_flag = 1 !Set flag to re-write foasetup
1328  INPUT "Enter the new correction wavelength: ", Wavelen
1330  IF Wavelen < 800 OR Wavelen > 1600 THEN
1332  BEEP
1334  DISP "SYSTEM DATA -- Correction wavelengths must be between 800 and 1600."
1336  OFF ERROR
1338  GOTO Sys_modify
1340  END IF
1350  BEEP
1352  INPUT "Enter the new Z-axis correction: ",Zaxis
1354
1356  | Find the place to put the new correction values
1358
1360  FOR Index=0 TO Num_focus
1362  IF Focus(Index,0)>Wavelen THEN GOTO Add
1364  IF Focus(Index,0)=Wavelen THEN GOTO Replace
1366  NEXT Index
1368
1370  | Now make room for the new value by shifting up by 1
1372
1374  Add:FOR IX=Num_focus+1 TO Index+1 STEP -1
1376  Focus(IX,0)=Focus(IX-1,0)
1378  Focus(IX,1)=Focus(IX-1,1)
1380  Focus(IX,2)=Focus(IX-1,2)
1382  Focus(IX,3)=Focus(IX-1,3)
1384  NEXT IX
1386
1388  | Add the new value and update the count (num_focus)
1390
1392  Focus(IX,0)=Wavelen
1394  Focus(IX,1)=Xaxis
1396  Focus(IX,2)=Yaxis
1398  Focus(IX,3)=Zaxis
1400  Num_focus=Num_focus+1
1402  GOTO Mod_done
1404
1406  | If the specified wavelength already exists, replace it
1408
1410  Replace:Focus(Index,0)=Wavelen
1412  Focus(Index,1)=Xaxis
1414  Focus(Index,2)=Yaxis
1416  Focus(Index,3)=Zaxis
1418  Mod_done:OFF ERROR
1420  RETURN
1422
1424  | This code is executed if the set-up file does not exist
1426  | and the user wants to create one.
1428
1430  Sys_create:GOSUB Clr_screen
1432  OFF ERROR
1434  ON ERROR GOSUB Input_error
1436  OFF KEY
1438  Write_flag=2  | Set flag for creating a new foasetup
1440  BEEP
1442  INPUT "Enter the machine's serial number: ",Serial_num$
1444  BEEP
1446  INPUT "Enter the low wavelength range limit: ",Low_wave
1448  BEEP
1450  INPUT "Enter the high wavelength range limit: ",High_wave
1452  BEEP
1454  INPUT "Enter the detector switch wavelength: ",Det_switch
1456  BEEP
1458  INPUT "Enter the laser 1 wavelength: ",Laser(0)
1460  BEEP
1462  INPUT "Enter the laser 2 wavelength: ",Laser(1)
1464  BEEP
1466  INPUT "Enter the laser 3 wavelength: ",Laser(2)
1468  BEEP
1470  DISP "Does the system have a filter wheel or monochromator?"
1472  ON KEY 5 LABEL Keys$$(18) GOTO Set_flag
FOR Index=0 TO Num_focus
    PRINT TABXY(1,18);"Enter the wavelength for correction point ";Index+1
    INPUT Focus(Index,0)
    GOSUB Clr_screen
    PRINT TABXY(1,18);"Enter the X-axis correction for point ";Index+1
    INPUT Focus(Index,1)
    GOSUB Clr_screen
    PRINT TABXY(1,18);"Enter the Y-axis correction for point ";Index+1
    INPUT Focus(Index,2)
    GOSUB Clr_screen
    PRINT TABXY(1,18);"Enter the Z-axis correction for point ";Index+1
    INPUT Focus(Index,3)
    GOSUB Clr_screen
NEXT Index
No:OFF KEY
OFF ERROR
ON ERROR GOTO File_error
DISP ""
RETURN
Hang_here:GOTO Hang_here
Hang_here:OFF KEY
GOTO Hang_here
SELECT ERRN
CASE 32
  BEEP
CASE ELSE
  BEEP
  DISP "INVALID CHARACTERS ENTERED: SYSTEMDATA -- 
  ON KEY 5 LABEL "PROCEED" GOTO Gohead1
  Hang_here:GOTO Hang_here
  Gotohead1:OFF KEY
  END SELECT
  RETURN
File_error:OFF ERROR
DISP "The set-up file doesn't exist on this disk. Do you want to create one?"
ON KEY 5 LABEL "YES" GOTO Set_create
ON KEY 5 LABEL "NO" GOTO Don_t_store
ON KEY 8 LABEL "STORE CURRENT" GOTO Store_it
Wait_create:GOTO Wait_create
Don_t_store:OFF KEY
Write_flag=0
GOTO Sys_done
Set_create:OFF KEY
Write_flag=3
GOTO Sys_done
Store_it:OFF KEY
Write_flag=2
CASE 54
PURGE "foasetup"
CASE 55
BEEP
DISP "The directory has overflowed. Use a different disk."
ON KEY 5 LABEL "READY" GOTO Disk_change
Wait_disk:GOTO Wait_disk
Disk_change:OFF KEY
BEEP
DISP "SYSTEMDATA --&ERRM$
Dead_in_h20:GOTO Dead_in_h20
END SELECT
Sys_done:ON ERROR GOTO File_error
OFF KEY
IF Write_flag>0 THEN
IF Write_flag=3 THEN GOSUB Sys_create
IF Write_flag=1 THEN
PURGE Filename$
END IF
CREATE ASCII Filename$,27
ASSIGN @Setupfile TO Filename$
OUTPUT @Setupfile:"VERSION 2.1"
OUTPUT @Setupfile:Serial_num$
OUTPUT @Setupfile:Low_wave,High_wave
OUTPUT @Setupfile:Laser(*)
OUTPUT @Setupfile:Filter_flag
OUTPUT @Setupfile:Filter(*)
OUTPUT @Setupfile:Num_focus
OUTPUT @Setupfile:F,(*)
OUTPUT @Setupfile:Cutoff
OUTPUT @Setupfile:Pin_x,Pin_y,Pin_z
OUTPUT @Setupfile:Inx_step,Iny_step
OUTPUT @Setupfile:Outx_step,Outy_step
OUTPUT @Setupfile:Farfield_step
OUTPUT @Setupfile:Lnoise
OUTPUT @Setupfile:Det_switch
ASSIGN @Setupfile TO *
END IF
GOSUB Clr_screen
DISP ""
SUB Timeset(OPTIONAL Timedate$)
SET TIME/DATE MODULE

VERSION 2.1
*Dim Month$(1:12)(3)
*Read Month$(*)
*If NPAR#1 Then
*Hours=VAL(Timedate$)
*Minutes=VAL(Timedate$(POS(Timedate$,:)+1:2))
*Month=VAL(Timedate$(POS(Timedate$," ")+1:2))
*Timedate$=Timedate$(POS(Timedate$," ")+1,LEN(Timedate$))
*Day=VAL(Timedate$)
*Year=VAL(Timedate$(" ")+1,LEN(Timedate$))
*Goto Set_time
*End If
*Gosub Clr_screen  'Clear screen
*Retry:Print Tabxy(1,16);"Please enter the current time. Enter the hours and
*minutes."
*Print Tabxy(1,17);"(in 24-hour format) separated by a colon. Example: 1
*3:05"
*Beep
*Input Hours$
*Gosub Clr_screen
*If pos(Hours$,":")=0 Then
*Beep
*Print Tabxy(1,16);"Please enter the minutes (0-59):"
*Input Minutes$
*Hours=Val(Hours$)
*Minutes=Val(Minutes$)
*Else
*Enter Hours$ Using "K,K";Hours,Minutes
*End If
*Gosub Clr_screen  'Clear screen again
*Get_month:Print Tabxy(1,16);"Please enter the month as a three-letter abbre
*viation."
*Print Tabxy(1,17);"(JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NO
*V, DEC)"
*Beep
*Input Mon$
*For I=1 To 3  'Change lower case to upper case in month
*Mon$(I:1)=Chr$(Binand(num(Mon$(I:1)),Bincmp(32)))
*Next I
*Month=0
*For I=1 To 12  'Look for the month in month$
*If pos(Mon$,Month$(I)) Then Month=I
*Next I
*If Month=0 Then
*Beep
*Print Tabxy(1,10);"TIMESET' You have entered an invalid month, please
*try again."
*Goto Get_month
*End If
*Gosub Clr_screen
*Beep
*Print Tabxy(1,16);"Please enter the day of the month (1-31):"
*Input Day
*Beep
*Print Tabxy(1,16);"Please enter the last two digits of the year:"
*Input Year
*Set_time:If Month>2 Then
*Month=Month-3
*Else
*Month=Month+9
*Year=Year-1
*End If
*Year=Year+1900
IF Julian<2.08662912E+11 OR Julian>=2.14325222E+11 THEN Range_err

SET TIMEDATE Julian

GOTO Done$

Range_err:BEEP

GOSUB Clr_screen

PRINT "TIMESET -- The time or date entered was out of range. Please try again."

GOTO Retry

Syntax_err:BEEP

GOSUB Clr_screen

PRINT TABXY(1,10);"TIMESET -- Syntax error. Please try again."

GOTO Retry

Clr_screen:OUTPUT KBD USING ";","K","K" Clear screen without scroll

RETURN

Done:OFF ERROR

GOSUB Clr_screen

DISP "" Clear error messages

SUBEND

DEF FNTimedate$

*****************************************************************************

GET CURRENT TIME/DATE MODULE

*****************************************************************************

DATA JAN,FEB,MAR,APR,MAY,JUN,JUL,AUG,SEP,OCT,NOV,DEC

DIM Month$((1:12)(3]

READ Month$$(*)

Compute the current hours, minutes, and seconds

Time_now=INT(TIMEDATE) MOD 86400

Hours=Time_now DIV 3600

Minutes=Time_now MOD 3600 DIV 60

Seconds=Time_now MOD 60

Find/Compute the current date

Julian=TIMEDATE DIV 86400-1721149

Year=(4*Julian-1) DIV 146097

Julian=(4*Julian-1) MOD 146097

Day=Julian DIV 4

Julian=(4*Day+3) DIV 1461

Day=(4*Day+3) MOD 1461

Day=(Day-4) DIV 4

Month=(5*Day-3) DIV 153

Day=(5*Day-3) MOD 153

Day=(Day+5) DIV 5

Year=(100*Year+Julian)-1900

IF Month<10 THEN

Month=Month+3

ELSE

Month=Month-9

Year=Year+1

END IF

Timedate$=VAL$(Day)"-"&Month$(Month)&"-"

Year$=VAL$(Year)

IF Year<=0 THEN Year$="0"&Year$

Hours$=VAL$(Hours)

IF Hours<10 THEN Hours$="0"&Hours$

Minutes$=VAL$(Minutes)

IF Minutes<10 THEN Minutes$="0"&Minutes$
RETURN LOCKTIME
FNEND

SUB Logtime(OPTIONAL Clrflag)

<table>
<thead>
<tr>
<th>LOG TIME AND DATE MODULE</th>
</tr>
</thead>
</table>

2004  
2006  
2008  
2010  
2012  
2014  
2016  
2018  
2020  
2022  
2024  
2026  
2028  
2030  
2032  
2034  
2036  
2038  
2040  
2042  
2044  
2046  
2048  
2050  
2052  
2054  
2056  
2058  
2060  
2062  
2064  
2066  
2068  
2070  
2072  
2074  
2076  
2078  
2080  
2082  
2084  
2086  
2088  
2090  
2092  
2094  
2096  
2098  
2100  
2102  
2104  
2106  
2108  
2110  

SUB Archive(OPTIONAL Files)

DIM Filename$(10), Temp(256,1)

INTEGER Index, Log_index, Log_flag(6)

FOR Index=0 TO 6
    Log_flag(Index)=0
    Log flags correspond to Fiber test #’s
NEXT Index

Compute the required file size, and set log flags

Numrec=8 ! Initial space for file header

IF Fiber_id$=Specatt_id$[1, LEN(Fiber_id$)] THEN
    Log_flag(1)=1 ! Log spectral attenuation data
    Numrec=Numrec+400
END IF

IF Fiber_id$=Dma_id$[1, LEN(Fiber_id$)] THEN
    Log_flag(2)=1 ! Log DMA data, 200 for wavelengths
    Numrec=Numrec+200+(200*Dmarundata(2,0)) ! Variable for signal data
END IF

IF Fiber_id$=FField_id$[1, LEN(Fiber_id$)] THEN
    Log_flag(3)=1 ! Log far-field data
    Numrec=Numrec+440 ! Fibertest 4 data stored here also
END IF

IF Fiber_id$=NField_id$[1, LEN(Fiber_id$)] THEN
    Log_flag(5)=1 ! Log near-field data
    Numrec=Numrec+240
END IF

IF Numrec<>8 THEN
    BEEP
    DISP "ARCHIVE -- There is no data in memory with the current fiber I.D"
    HALT 2
Got the file name from the user and open the file.

IF NPAR>8 THEN
  File_name=File$
  GOTO Open_file
ELSE
  GOTO Get_name
END IF

Get_name:
PRINT TABXY(1,16);"Please put the disk on which the data is to be archived in the right hand drive."
PRINT TABXY(1,17);"Then enter archive data file name (10 letters max). Press PROCEED when ready."
ON KEY $ LABEL "PROCEED" GOTO Open_file
WAIT 2
INPUT File_name$
Hang_man: GOTO Hang_man
Open_file: ON ERROR GOTO File_err
CREATE DAT File_name$Arcdrive$,Numrec,8
ASSIGN @Archive TO File_name$Arcdrive$
OUTPUT KBD USING "*,K","K"
PRINT TABXY(10,1);CHR*(129)&"FOA-2000 measurement data archive utility. &CHR$(128)
PRINT TABXY(1,3);"Archiving data for fiber: ";Fiber_id$
OUTPUT @Archive;FNTimedate$
WAIT 2
Select data to be written from the log flags
IF Log_flag(1)=1 THEN
  OUTPUT @Archive;"SPECATTEN"
  OUTPUT @Archive;Specatt_id$
  OUTPUT @Archive;Specattdata(*)
END IF
IF Log_flag(2)=1 THEN
  OUTPUT @Archive;"ODA"
  OUTPUT @Archive;Dma_id$
  OUTPUT @Archive;Dmaattendata(*)
END IF
IF Log_flag(3)=1 THEN
  OUTPUT @Archive;"FFIELD"
  OUTPUT @Archive;Field_id$
  OUTPUT @Archive;Field(*)
END IF
IF Log_flag(5)=1 THEN
  OUTPUT @Archive;"NFIELD"
  OUTPUT @Archive;Nfield_id$
  OUTPUT @Archive;Nearfield(*)
END IF
ASSIGN @Archive TO *
GOTO Done
File_err: SELECT ERRN
CASE 54 = File name already exists.
  DISP "File ";File_name$;" already exists. Do you want to delete it or change the name?"
ON KEY 5 LABEL "YES" GOTO Yes
ON KEY 6 LABEL "NO" GOTO Done
ON KEY 7 LABEL "CHANGE" GOTO Chng_nm
Wait_1: GOTO Wait_1
CASE 53
BEEP
DISP "Unknown file name. Must be 10 characters or less with no spaces"
CASE 64
BEEP
DISP "The archive disk is full. Replace with a new disk. Initialize if necessary."
ON KEY S LABEL "PROCEED" GOTO New_disk
Wait_full:GOTO Wait_full
New_disk:OFF KEY
DISP ""
GOTO Open_file
CASE 80
BEEP
DISP "The "&Arcdrive$&" disk drive is empty. Please insert the archive disk."
ON KEY S LABEL "PROCEED" GOTO Disk_ready
Wait_2:GOTO Wait_2
Disk_ready:DISP ""
OFF KEY
GOTO Open_file
CASE ELSE
DISP "ARCHIVE -- HP Error Number "&VAL$(ERRN)
GOTO Done
END SELECT
Yes:OFF KEY
DISP ""
PURGE Filename$&Arcdrive$
GOTO Open_file
Chng_nm:OFF KEY
GOTO Get_name
Done:OUTPUT KBD USING "$","K"
Erase the screen
GOTO Open_file
Chng_nm:OFF KEY
GOTO Get_name
Done:OUTPUT KBD USING "$","K"
Erase the screen
SUB Retrieve(OPTIONAL File$);
RETRIEVE ARCHIVED MEASUREMENT DATA MODULE
VERSION 2.1P
DIM Filename$(30),Data_type$(70)
INTEGER Index,Jindex
PRINT CHR$(12)
Get the file name from the user
IF NPAR>0 THEN
Filename$=File$
GOTO Open_file
ELSE
GOTO Get_file
END IF
PRINT TABXY(1,16);"Please put the disk containing the archived file in the right-hand drive."
PRINT TABXY(1,17);"Then enter the name of the archived file. Press PROC
GOTO Hang.glrl

Ovn-file:ON ERROR GOTO File_err

ASSIGN @Archive TO Filenam$&Arcdrive$

| Read the archive time and date from the first line of the file.

| ENTER @Archive:Archive_date$

| OUTPUT KBD USING "$","K" !Erase screen

| PRINT TABXY(1,5):"Retrieving data archived on: "&Archive_date$

| PRINT TABXY(1,7):"This archive file contains the following data:

| PRINT

| Read the data type header. If it's SPECATTEN, read the following array into Specattdata(*). If not, check the other data types.

| ENTER @Archive:Data_type$

| IF Data_type$="SPECATTEN" THEN

| ENTER @Archive:Specatt_id$

| ENTER @Archive:Specattdata(*)

| PRINT "Spectral Attenuation Data"

| PRINT " for fiber: "&Specatt_id$

| END IF

| Read DMA data (if any).

| IF Data_type$="DMA" THEN

| ENTER @Archive:Dma_id$

| ENTER @Archive:Dmaattenda(*)

| PRINT "Differential Modal Attenuation Data"

| PRINT " for fiber: "&Dma_id$

| END IF

| Read Far-field data (if any).

| IF Data_type$="FFIELD" THEN

| ENTER @Archive:Nfield_id$

| ENTER @Archive:Nearfield(*)

| PRINT "Near field data"

| PRINT " for fiber: "&Nfield_id$

| END IF

| Read Near-field data (if any).

| IF Data_type$="NFIELD" THEN

| ENTER @Archive:Nfield_id$

| ENTER @Archive:Nearfield(*)

| PRINT "Near field data"

| END IF

| ASSIGN @Archive TO * !Close the file

| GOTO Done

| File_err:IF ERR$=59 THEN !Error 59=End of file reached.

| ASSIGN @Archive TO *

| GOTO Done !Close the file

| We're done.

| END IF

| IF ERR$=80 THEN

| DISP "No disk in right hand drive, please insert and try again."

| GOTO Get_file

| END IF

| DISP "File name contains unrecognizable characters."
DISP "The archive file "&Filename$" isn't on this disl. Do you want to try again?"
ON KEY 5 LABEL " YES" GOTO Yes
ON KEY 6 LABEL " NO" GOTO Quit
Wait here: GOTO Wait_here
Yes: !
OFF KEY
OUTPUT KBD USING ",K","K" !Erase screen
CAT Arcdrive$
GOTO Get_file
END IF
DISP "RETRIEVE -- HP Error Number ":&VAL$(ERRN)
ON KEY 5 LABEL "RETURN" GOTO Yes
Dead_in_h20: GOTO Dead_in_h20
Unspaced: OFF KEY
CALL Cleardisplay
GOTO Get_file
Done: IF NPAR>0 THEN GOTO Quit
ON KEY 5 LABEL "CONTINUE" GOTO Quit
Wait_done: GOTO Wait_done
Quit: OUTPUT KBD USING ",K","K"
OFF KEY
SUBEND
SUB Zcenter
Z-AXIS MOTOR CENTERING MODULE
VERSION 2.1
\\**** NOTE ****
\\The FOA-2000 commands used in this module are not documented in
\\the FOA-2000 manual and should be used only under direction of
\\Photon Kinetics.
\\CALL F2000send("ALIGN INZ COUPL 3000 DARK",1) !Find edge of INZ sensor
\\CALL F2000send("INZ ZER -900 GOTO INZ ZER",1) !Back up and stop
\\CALL F2000send("OUTZ COUPL 3000 DARK",1) !Find edge of OUTZ sensor
\\CALL F2000send("OUTZ ZER -900 GOTO OUTZ ZER",1) !Back up and stop
\\SUBEND
\\SUB Rundisplay(Message$)
\\IN PROCESS DISPLAY MODULE
\\VERSION 2.1
\\GINIT
\\GCLEAR
\\GRAPHICS ON
\\MOVE 0,30
\\CSIZE 5,57
\\LABEL Message$
\\SUBEND
\\SUB Cleardisplay
\\CLEAR DISPLAY - clears both alpha and graphics
\\OFF KEY
\\DISP " " !Clears labels from bottom of screen
\\CLEAR HEADER
\\OUTPUT KBD USING ",K","K" !Clears alphanumeric characters
SUB F2000send(Mess$,OPTIONAL Wait_flag)

SEND COMMANDS TO FOA-2000 MODULE

COM \iopath$/ @FoA2000,0Egg5205,0TeK7854,0Bncdelay,Printer_add

DIM Message$(80)

INTEGER Statbyt,Busybit

INTEGER Posn

Busybit=4

Message$=Mess$

DISP "FOA-2000: ";"Message$

Busy:=Statbyt=SPOLL(@Foa2000)

IF BIT(Statbyt,Busybit)=1 THEN GOTO Busy

IF Err_flag THEN DISP "FOA-2000: ";"Message$

OUTPUT @Foa2000 USING "K";"Message$

Previous$=Message$

IF Statbyt,99 AND Statbyt<100 THEN

Err_flag=1

END IF

RETURN

OORTE'DISP ""SUBEND

SYSTEM PRESET MODULE

COM \iopath$/ @FoA2000,0Egg5205,0TeK7854,0Bncdelay,Printer_add

DIM Message$(80)

INTEGER Statbyt,Busybit

INTEGER Posn

Busybit=4

Message$=Mess$

DISP "FOA-2000: ";"Message$

Busy:=Statbyt=SPOLL(@Foa2000)

IF BIT(Statbyt,Busybit)=1 THEN GOTO Busy

IF Err_flag THEN DISP "FOA-2000: ";"Message$

OUTPUT @Foa2000 USING "K";"Message$

Previous$=Message$

IF Statbyt,99 THEN

PRINT TABXY(1,15);"FOA-2000 ERROR NUMBER: "&VAL$(Statbyt) &" &Previous$.

PRINT TABXY(1,15);"Motor error. Probably caused by fiber misalignment or a bad fiber end."

PRINT TABXY(1,17);"First try focussing the fiber end on the screen, and press PROCEED."

CALL F2000send("ALIGN")

ELSE

PRINT TABXY(1,15);"FOA-2000 ERROR NUMBER: "&VAL$(Statbyt) &" &Previous$.

END IF

LOCAL @Foa2000

ON KEY 5 LABEL "PROCEED" GOTO Proceed

Wait_here:GOTO Wait_here

Proceed:=DISP "FOA2000: ";"Previous$

OUTPUT @Foa2000 USING "K";"Previous$

WAIT .01

Serial poll the instrument

IF BIT(Statbyt,Busybit)=1 THEN Busy:

GOTO Err_chk

Check for errors once more

END IF

RETURN

Done:=DISP ""

SUBEND

SYSTEM PRESET MODULE

COM \iopath$/ @FoA2000,0Egg5205,0TeK7854,0Bncdelay,Printer_add

DIM Message$(80)

INTEGER Statbyt,Busybit

INTEGER Posn

Busybit=4

Message$=Mess$

DISP "FOA-2000: ";"Message$

Busy:=Statbyt=SPOLL(@Foa2000)

IF BIT(Statbyt,Busybit)=1 THEN GOTO Busy

IF Err_flag THEN DISP "FOA-2000: ";"Message$

OUTPUT @Foa2000 USING "K";"Message$

Previous$=Message$

IF Statbyt,99 THEN

PRINT TABXY(1,15);"FOA-2000 ERROR NUMBER: "&VAL$(Statbyt) &" &Previous$.

PRINT TABXY(1,15);"Motor error. Probably caused by fiber misalignment or a bad fiber end."

PRINT TABXY(1,17);"First try focussing the fiber end on the screen, and press PROCEED."

CALL F2000send("ALIGN")

ELSE

PRINT TABXY(1,15);"FOA-2000 ERROR NUMBER: "&VAL$(Statbyt) &" &Previous$.

END IF

LOCAL @Foa2000

ON KEY 5 LABEL "PROCEED" GOTO Proceed

Wait_here:GOTO Wait_here

Proceed:=DISP "FOA2000: ";"Previous$

OUTPUT @Foa2000 USING "K";"Previous$

WAIT .01

Serial poll the instrument

IF BIT(Statbyt,Busybit)=1 THEN Busy:

GOTO Err_chk

Check for errors once more

END IF

RETURN

Done:=DISP ""

SUBEND
2750 OUTPUT KBD USING "$K",K"     ! Clear alpha screen
2752 CALL Rundisplay("    Initializing system equipment.")
2754 CALL F2000send("3 ATTENUAT CHOP-ON LAMP-ON LED-ON GERMAIN")
2756 
2758 Center the FOA-2000 focus motors.
2760 
2762 PRINT TABXY(1,16);"Centering the FOA-2000 focus motors."
2764 CALL Zcenter
2766 
2768 Now wait for the operator to confirm warm-up
2770 
2772 BEEP
2774 PRINT TABXY(1,16);"Please check that all equipment is ON. The FOA-2000 must be allowed to warm"
2776 PRINT TABXY(1,17);"up for 5 minutes before proceeding. Press the PROCEEDkey (FS) when ready."
2778 ON KEY 5 LABEL "PROCEED" GOTO Proceed
2780 Here:GOTO Here   !Wait for them to press the key
2782 Proceed:OFF KEY
2784 OUTPUT KBD USING "$K","K"
2786 
2788 CALL F2000send("LED LED-ON ILLUMIN VOUT TARGET-OUT SPOT-OUT FF-OUT 3 ATT
ENUAT",1)
2790 
2792 Check chopper operation by looking at reference status on EGG
2794 
2796 Eggstatus=SPOLL(@Egg5205)
2798 IF BIT(Eggstatus,3)=1 THEN
2800 Retry:OUTPUT KBD USING "$K","K"    !Clean up screen from error
2802 OFF KEY
2804 PRINT TABXY(1,16);"Waiting for the FOA-2000 chopper to stabilize."
2806 CALL F2000send("CHOP-OFF CHOP-ON",1)
2808 Starttime=TIMEDATE
2810 Wait_loop:WAIT 2
2812 Eggstatus=SPOLL(@Egg5205)
2814 IF BIT(Eggstatus,3)=0 THEN GOTO Chop_ready
2816 IF TIMEDATE-Starttime>30 THEN
2818 BEEP
2820 DISP "The FOA-2000 Chopper is inoperative, or the lock-in amplifier reference channel is disconnected."
2822 ON KEY 5 LABEL "Retry" GOTO Retry
2824 ON KEY 6 LABEL "Stop" GOTO Quit
2826 Wait_key:GOTO Wait_key
2828 Quit:OFF KEY
2830 Dead!:GOTO Dead!
2832 END IF
2834 GOTO Wait_loop
2836 Chop_ready:WAIT 10    ! Wait 10 more seconds
2838 END IF              ! Skip the whole thing if REF LOW bit is not set
2840 
2842 Initialize EGG 5205/7 settings +, sending selected device clear (SDC).
2844 
2846 PRINT TABXY(1,16),"Setting up the EGG5207 Lock-in Voltmeter."
2848 CLEAR @Egg5205
2850 
2852 Set the EGG5207 phase (twice, for assured precision)
2854 
2856 CALL E5205comm("A2 1")
2858 CALL E5205comm("A2 1")
2860 
2862 Set the EGG5207 to a known range
2864
In this case, BE SURE THAT THE LAMP IS TURNED ON before proceeding.

To skip this step, press the SKIP key (f6).

ON KEY S LABEL "PROCEED" GOTO Mono_cal
ON KEY I LABEL "SKIP" GOTO Skipped
Hang_out: GOTO Hang_out

Mono_cal: OFF KEY
OUTPUT KBD USING ",K","K"
CALL F2000send("O ATTENUAT",1)
IF Filter_flag=1 THEN GOTO Done
CALL F2000send("O FILTER LAMP SOURCES 1TO1 SEEK WAV COUPL",1)
LOCAL @Foa2000
PRINT TABXY(1,16);"Please adjust the monochromator wavelength for the zero-point calibration (light"
PRINT TABXY(1,17);"centered on the launch spot, approaching it using a clockwise knob rotation)."

BEEP
ON KEY S LABEL "PROCEED" GOTO Wave_cal
Wait3:GOTO Wait3
Wave_cal: CALL F2000send("WAVE0")
CALL F2000send("WAVE0")
Skipped: OFF KEY
IF Alignment(0)=0 THEN CALL Fibertype !Query for fibertype
Done:OUTPUT KBD USING ",K","K"
CALL F2000send("ALIGN",1) !Leave the system in alignment set up
LOCAL @Foa2000 !Also leave the control panel in local mode
CALL Cleardisplay

SUBEND

SUB E5205comm(Message$,OPTIONAL Value)
!*****************************************************************************************
EGGS205 COMMUNICATION MODULE VERSION 2.1P
*******************************************************************************
COM /Iopaths/ @Foa2000,@Egg5205,@Tek7854,@Bncdelay,Printer_add
INTEGER Eggstatus
DISP "EGGS205: ";Message$
Start=TIMEDATE

Send the command or query to the S205

Busy:GOSUB Poll_egg !Serial poll the EGGS205
IF TIMEDATE-Start>5 THEN GOTO Timeout !Report timeout
IF NOT BIT(Eggstatus,0) THEN GOTO Busy
OUTPUT @Egg5205 USING "K";Message$
!It's ready, send message

Take in a response from the S205 if one is indicated; Wait for
"command complete" and "settled" before returning.

Wait_done:GOSUB Poll_egg
IF BIT(Eggstatus,7) THEN ENTER @Egg5205;Value$
IF NPAR>1 THEN
Value=Value$
END IF
IF BINTD(Eggstatus,33)<33 THEN GOTO Wait_done
DISP " 
GOTO Done
Poll_egg:WAIT .01
Eggstatus=SPOLL(@Egg5205) !Serial poll
RETURN
DEF FNUltimate(Accuracy)

'**************************************************************************
<p>| EGG5205 VOLTMETER READING MODULE                                        |</p>
<table>
<thead>
<tr>
<th>VERSION 2.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>COM /Egg5205/ Scales(*),Settle,INTEGER Num_avet,Range</td>
</tr>
<tr>
<td>COM /Syscalc/ Pin_x,Pin_y,Pin_z,Inx_step,Iny_step,Outx_step,Outy_step,</td>
</tr>
<tr>
<td>Field_step,Noislevlern</td>
</tr>
<tr>
<td>COM /Iopaths/ @Foa2000,8Egg5205,8Tek496p,8Tek70S4,Printer_add</td>
</tr>
<tr>
<td>DIM Oldreading(31)</td>
</tr>
<tr>
<td>INTEGER Index,I,Down_count,Num_readings,Referlow,Overload</td>
</tr>
<tr>
<td>Lowest_range=12                                                         !Lowest allowed EGG5205 scale =500uv</td>
</tr>
</tbody>
</table>

IF Accuracy<>0 THEN
    Call ES20ScomW(S" ,Rangeread)
    Range=INT(Rangeread)
END IF

IF Accuracy=0, the number of averages and range should not be adjusted.
If Accuracy<>0, compute the number of averages required to achieve the
requested accuracy. Accuracy is expressed in dB. Use TEMP to avoid
INTEGER overflow.

Restart: IF Accuracy<>0 THEN
    Error=.23*Accuracy
    Temp=(Noiselevel/(Error*400*Scales(Range)))^2
    IF Temp<6 THEN Temp=6  !Minimum # of averages = 6
    IF Temp>30 THEN Temp=30!Maximum # of averages = 30
    Num_avet=Temp
    Settle=Error*400*Scales(Range)  !Settling requirement
END IF

Sum=0
Sum_squares=0
Num_readings=0
FOR Index=0 TO Num_avet
    Oldreading(Index)=0
NEXT Index
Index=0
Res_limit=Scales(Range)  !Resolution limit is 1 LSB

Get a voltage reading
END IF
3128 IF Accuracy<>0 THEN 'DON'T down range if accuracy=0
3130 IF ABS(Reading)<400 AND Range<Lowest_range THEN,
3132 GOSUB Down_range
3134 GOTO Restart 'Start over on averages
3136 END IF
3138 END IF
3140 IF ABS(Reading)>2000 THEN 'We can always try to up-range
3142 GOSUB Up_range
3144 GOTO Restart 'Start over on averages
3146 END IF
3148 I This reading is within the range limits, so scale it into volts
3150 I before adding it to the running sum and computing standard
3152 I deviation.
3154 I
3155 Reading=Reading*Scales(Range)
3156 I
3158 I A running sum and sum of squares is kept of the number of most
3160 I current readings specified by num_aver. Each time a new reading
3162 I is added to the running sum the oldest reading is removed from
3164 I the sum so that the sum always reflects the most current readings.
3166 I
3168 Sum=Sum+Reading-Oldreading(Index) 'Update the sums
3170 Sum_squares=Sum_squares+(Reading^2)-Oldreading(Index)^2
3172 Oldreading(Index)=Reading 'Replace old reading with new one
3174 Index=(Index+1) MOD Num_aver 'And update oldreadings index
3176 Num_readings=Num_readings+1 'Count the new reading
3178 I If we have acquired at least num_aver readings, compute the standard
3180 I deviation of the last num_aver readings and compare it to the noise
3182 I limit and resolution limit. If the result is inside these limits,
3184 I the EGGS205 has settled, so return the average of the readings.
3186 I
3188 IF Num_readings>=Num_aver THEN 'If acquired enough, check noise
3190 Noise=SQR(ABS(Sum_squares-(Sum)^2/Num_aver)/Num_aver)
3192 IF Noise<Noiselevel OR Noise<Res_limit OR Noise<Settle OR TTMEDATE-TI
3194 THEN
3196 Result=Sum/Num_aver 'If noise is within limits, return
3198 GOTO Done 'the average of the readings.
3200 END IF
3202 END IF
3204 GOTO Acquire 'If not enough averages or too much
3206 I noise, go get another reading
3208 I
3210 Poll the EGGS205 and break its status down into 2 conditions:
3212 Reference low and Overload These conditions are returned to
3214 as separate variables with a value of 1 if the condition is
3216 true or 0 if it is false.
3218 I
3220 Pollegg:Eggstatus=SPOLL(@Egg5205)
3222 Referlow=Bit(Eggstatus,3)
3224 Overload=Bit(Eggstatus,4)
3226 RETURN
3228 I This subroutine increments the EGGS205 range when the reading is
3230 greater than 2000 or when overload status occurs.
3232 I
3234 Up_range:IF Range=0 THEN 'We're already at highest range
3236 BEEP
3238 DISP "VOLTMETER -- EGGS205 is overrange on highest range."
3240 ON KEY 5 LABEL "PROCEED" GOTO Restart
3242 Wait8:GOTO Wait8
3244 "Wait for a key press key"
3258     Range=Range-1  ! (only 1 step if no steps back
3260     ELSE
3262     Range=Range-1
3264     END IF
3266     IF Range<0 THEN Range=0
3268     CALL ES20Scomm("S \&VAL$(Range))  ! Set the EGG to the new range
3270     WAIT 1  ! Time for EGG5205 transient
3272     RETURN
3274
3276     ! This subroutine decrements the EGG5205 sensitivity to achieve readings greater than 400.
3278
3280     Down_range:Vmagnitude=ABS(Reading*Scales(Range))
3282     FOR I=1 TO 5  ! Max allowed range change=5 steps
3284     Range=Range+1
3286     IF Vmagnitude>400*Scales(Range) THEN Set_down
3288     NEXT I
3290     Set_down: IF Range>Lowest_range THEN Range=Lowest_range
3292     CALL ES20Scomm("S \&VAL$(Range))
3294     WAIT 1  ! Allow recovery time
3296     RETURN
3298
3300
3302     ! Return the average of the readings to the caller
3304
3306     Done: RETURN Result
3308     FNEND
3310
3312
3314     SUB Setscale(Accuracy,Maxvolts)
3316     !---------------------------------------------------------------
3318     ! SET EGG5205 RANGE MODULE  VERSION 2.1
3320     !---------------------------------------------------------------
3322     COM /Egg5205/ Scales(*),Settle,INTEGER Num_aver,Range
3324     COM /Syscal/ Pin_x,Pin_y,Pin_z,Inx_step,Inystep,Outxstep,Outystep,Far
3326     fieldstep,Noiselevel
3328     FOR Index=14 TO 0 STEP -1  ! Figure out the appropriate range
3330     IF Scales(Index)*2000>=Maxvolts THEN GOTO Set-range
3332     NEXT Index
3334     Index=0  !Maxvolts is too big!
3336     PRINT TABXY(1,17)"SETSCALE -- The maximum voltage specified for"
3338     PRINT TABXY(1,18)"the EGG 520S is too large."
3340     Dead1:GOTO Dead1
3342     Set-range: Range=Index  ! Set the range
3344     CALL ES20Scomm("S \&VAL$(Range))
3346     Perror=.23*Accuracy  ! Convert dB to % error
3348     Num_aver=(Noiselevel/(Perror*400*Scales(Range)))/2
3350     IF Num_aver<6 THEN Num_aver=6  ! Minimum # of averages = 6
3352     IF Num_aver>30 THEN Num_aver=30  ! Maximum # of averages = 30
3354     Settle=Perror*400*Scales(Range)  ! Settling requirement
3356     SUBEND
3358
3360
3362     SUB Arraybuild(Instring$,Outarray(*),Arraylen)
3364     !-----------------------------------------------------------------------
3366     ! ARRAY BUILDER MODULE  VERSION 2.1
3368     !-----------------------------------------------------------------------
3370     FOR_flag=0  ! Set for loop flag = 0
3372     Arraylen=0  ! Set initial array length = 0
3374     Step_val=1  ! Set default step index value
3376     Temp$=""  ! Initialize temporary string
3378     ON ERROR GOTO Errorline
3380     END
3390 IF Index<LEN(Instring$)-1 THEN !Don't look if near the end
3392 IF Instring$[Index+2]=""TO" OR Instring$[Index+2]=""TO"" THEN !Look for "TO"
3394 GOSUB For_loop !Found a "TO"--go process it
3396 GOTO New_val
3398 END IF
3400 END IF
3402 !Check for "STEP" keyword
3406 !Don't look if near the end
3410 IF Instring$[Index+4]=""STEP" OR Instring$[Index+4]=""STEP"" THEN !STEP
3412 GOSUB Step_loop !Process the STEP
3414 GOTO Next_char
3416 END IF
3418 END IF
3420 !If the next character is not a number, decimal point (.) or minus sign,
3422 !it is a separator character, so figure out what to do about it. If the
3424 !next character is a number, . or -, just add it to the temp$.
3426 Value=NUM(Instring$[Index+1]) !Get the next char's value
3428 IF (Value<48 OR Value>57) AND Value<>32 AND Value<>46 AND Value<>45 THEN
3430 OUTarray(Arraylen)=VAL(Temp$) !It's just a regular value.
3432 Arraylen=Arraylen+1 !Increment the array length
3434 END IF
3436 !First, check to see if we have a FOR loop in process.
3440 New_val:SELECT For_flag
3442 CASE 1 !We've passed a "TO"
3444 GOSUB Load_start !Go load the starting index
3446 GOTO Next_char !And keep looking
3448 CASE 2 !This is the ending value
3450 GOSUB Load_end !Load ending index and run loop
3452 GOTO Next_char
3454 CASE 3 !A FOR loop with STEP value
3456 GOSUB Run_loop !Run the loop
3458 GOTO Next_char !And start checking again
3460 END SELECT
3462 OUTarray(Arraylen)=VAL(Temp$) !No FOR loop is in progress
3464 Arraylen=Arraylen+1 !It's just a regular value
3466 Temp$="" !Increment the array length
3468 END IF
3470 Next_char:NEXT Index !And clear the temporary string
3472 !Check the next input character
3474 !When we run out of characters in INSTRING$, check to see if
3476 !we have a FOR loop pending, or if it's just a regular value.
3478 !A "TO" with no ending value
3480 SELECT For_flag
3482 CASE 1 !That's a syntax error
3484 GOTO Syntax_err
3486 CASE 2 !A FOR loop ending with no STEP
3488 GOSUB Load_end !That's OK, go run the loop
3490 GOTO Done
3492 CASE 3 !A FOR loop with a STEP value
3494 GOSUB Run_loop !Load the STEP and run the loop
3496 GOTO Done
3498 END SELECT
3500 OUTarray(Arraylen)=VAL(Temp$) !It's just a last regular value.
3502 Arraylen=Arraylen+1 !Increment array length
3504 GOTO Done !And quit
Step_loop: IF For_flag<>2 THEN Syntax_err

End_index=VAL(Temp$)  !Load the ending index
Index=Index+3          !Point past the "STEP" keyword
For_flag=3
Temp$=""               !ready for the STEP value
RETURN

Load the starting value for a nn TO nn loop.

Load_start:Start_index=VAL(Temp$)  !Previous number is begin value
Temp$=""                          
For_flag=2                         !Next value will be ending value
RETURN

Load the ending value for a nn TO nn loop.

Load_end:End_index=VAL(Temp$)      !So put it in end index
FOR Value=Start_index TO End_index+Step_val/100 STEP Step_val
OUTarray(Arraylen)=Value          !Store the value in output array
Arraylen=Arraylen+1               !Increment output array pointer
NEXT Value
Temp$=""
For_flag=0                        !Reset the for flag
RETURN

Execute a nn TO nn loop

Run_loop:Step_val=VAL(Temp$)       !load step with this value
FOR Value=Start_index TO End_index+Step_val/100 STEP Step_val
OUTarray(Arraylen)=Value          !Load values in out array
Arraylen=Arraylen+1               !Increment output pointer
NEXT Value
Temp$=""
Step_val=1
For_flag=0                         !Reset the for flag
RETURN

Here's where we end up if an error has been trapped.
The only check is to see if the array has overflowed.
If it has, the number of points is calculated and the routine
is exited normally. If not, the error number is reported
and the program hangs.

Errorline: IF ERN = 17 THEN
SELECT For_flag
CASE 2,3
Arraylen=Arraylen+(End_index-Value)/Step_val
CASE 0
Arraylen=Arraylen+1
CASE ELSE
Arraylen=-1
END SELECT
END IF

ELSE
BEEP
PRINT TABXY(5,10);"ARRAYBUILD: Error #"&VAL$(ERN)&" has occurred."
PRINT TABXY(5,11);"Program idle."
END Select

DeadS: GOTO DeadS
END IF

Here's where we end up if we find a bad syntax.
SUB Fiberident

            COM /Fiber/ Fiber_ids,Fiber_len,Log_time$
DIM Id(80),Len$(80)

Im1:IMAGE ","Please enter the fiber identification: ",K
OUTPUT KBD USING Im1
BEEP
ENTER KBD USING Im1;Id$
IF LEN(Id$) THEN
Fiber_id=Id$
END IF

Im2:IMAGE ","Please enter the fiber length in meters (or zero): ",K
OUTPUT KBD USING Im2
BEEP
ENTER KBD USING Im2;Len$
IF LEN(Len$) THEN
Fiber_len=VAL(Len$)/1000
END IF
CALL Cleardisplay

SUBEND

SUB Fibertype(OPTIONAL Fiber_type)

COM /Align_param/ A(*)

Get_type: !
IF NPAR<1 THEN
INPUT "Please enter the fiber type (20, 50, 85, 100, or 150): ",Ftype
ELSE
Ftype=Fiber_type !If fiber_type argument is included, use it
END IF
SELECT Ftype
CASE 20
A(0)=1 !Queried to see if fibertype has been set (no=0)
A(1)=10 !Step size for rough alignment
A(2)=10 !Rough_dy
A(3)=100 !Rough_dz
A(4)=4 !Fine_dx !Step size for fine alignment
A(5)=4 !Fine_dy
A(6)=15 !Fine_dz

CASE 50
A(0)=1 !Queried to see if fibertype has been set (no=0)
A(1)=20 !Step size for rough alignment
A(2)=20 !Rough_dy
A(3)=200 !Rough_dz
A(4)=20 !Fine_dx !Step size for fine alignment
A(5)=20 !Fine_dy
A(6)=20 !Fine_dz

CASE 85
A(0)=1 !Queried to see if fibertype has been set (no=0)
A(1)=36 !Step size for rough alignment
A(2)=36 !Rough_dy
A(3)=272 !Rough_dz
A(4)=12 !Fine_dx !Step size for fine alignment
A(5)=12 !Fine_dy
A(6)=12 !Fine_dz

END
CASE 150
  A(0)=1
  "150 micron fiber diameter."
CASE ELSE
  BEEP
  GOTO Get_type
END SELECT
SUBEND

SUB Fiberload(String$)
  FIBER LOAD MODULE
  VERSION 2.1P

  COM /lopaths/ @Foa2000, @Egg5205, @Tek7854, @Bncdelay, Printer_add
  CALL F2000send("ALIGN",1)
  LOCAL @Foa2000
  OUTPUT KBD USING ",", "K"
  PRINT TABXY(1,10); String$
  BEEP
  ON KEY 5 LABEL "PROCEED" GOTO Proceed
  Here: GOTO Here
  Proceed: OFF KEY
  CALL F2000send("STAGE0",1)
  OUTPUT KBD USING ",", "K"
SUBEND

SUB Specwaves(String$)
  SPECTRAL ATTENUATION WAVELENGTHS MODULE
  VERSION 2.1

  COM /Wavelength/ Wavelength(*), Numsteps
  COM /Sysdata/ Serial_num$, Lasers(*), Filter_flag, Filter(*), Num_focus, Focus(*), Cutoff, Low_wave, High_wave, Det_switch

  The array builder parses the user's input string and builds a wavelength array.

  CALL Arraybuild(String$, Wavelength(*), Numsteps)
  IF Numsteps<=0 THEN
    BEEP
    "ARRAYBUILD got an error!"
    Disp "SPECTWAVES -- Bad command format. Program idle."
    Dead1: GOTO Dead1
  END IF
  IF Numsteps>350 THEN
    Check for more than 100 values
  END IF
3905 Numsteps=350  !Set number of steps to 350
3906 END IF
3910 !
3912 ! Now we just check to see if the values are all within the
3914 ! valid range.
3916 !
3918 FOR Index=0 TO Numsteps-1
3920 IF Wavelength(Index)<Low_wave THEN
3922 BEEP
3924 DISP "SPECWAVES -- A wavelength below "&VAL$(Low_wave)&" nm is speci
3925 fied. It will be set to "&VAL$(Low_wave)&" nm."
3926 WAIT 3
3928 Wavelength(Index)=Low_wave
3930 END IF
3932 IF Wavelength(Index)>High_wave THEN
3934 BEEP
3936 DISP "SPECWAVES -- A wavelength above "&VAL$(High_wave)&" nm is spec
3938 ified. It will be set to "&VAL$(High_wave)&" nm."
3938 WAIT 3
3940 Wavelength(Index)=High_wave
3942 END IF
3944 NEXT Index
3946 DISP "
3948 SUBEND
3950 !
3952 !
3954 SUB Setfocus(Wavelength)
3956 !+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
3958 ! SET FOA-2000 FOCUS CORRECTION MODULE
3960 !---------------------------------------------------------------
3962 COM /Sysdate/ Serial_num$,Lasers(*),Filter_flag,Filter(*),Num_focus,Focu
3964 s(*),Cutoff,Low_wave,High_wave,Det_switch
3966 INTEGER Index,WavelWave2 ,AindexCor
3968 DIM Cmd$[48]
3970 Axis$(9)="IN-X"
3972 Axis$(10)="IN-Y"
3974 Axis$(12)="IN-Z"
3976 ! Find the two entries in the focus correction table that are closest to
3978 ! the desired wavelength
3980 !
3982 !
3984 IF Num_focus<2 THEN SUBEXIT
3986 FOR Index=1 TO Num_focus-1
3988 IF Focus(Index,0)=Wavelength THEN GOTO Exit_loop
3990 NEXT Index
3992 !
3994 ! Next, get the correction value for the specified wavelength for
3996 ! each axis. If the specified wavelength was not found in the array,
3998 ! interpolate between the adjacent values to compute the correction
4000 ! value. This process is done for each axis (X, Y, and Z).
4002 !
4004 Exit_loop:FOR Aindex=1 TO 3
4006 GOSUB Get_cor
4008 Cmd$=Cmd$&VAL$(Cor_val)&"&Axis$(Aindex-1)&"
4010 NEXT Aindex
4012 CALL F2000send(Cmd$,1)
4014 GOTO Done
4016 Get_cor:Wave1=Focus(Index-1,0)
4018 Wave2=Focus(Index,0)
4020 Val1=Focus(Index-1,Aindex)
4022 Val2=Focus(Index,Aindex)
4024 Cor_val=(Val1+Val2)/2.0 "Will be granted the adjustment value for" #1
SUB Spectrun(OPTIONAL Spot$,Runmsg$)

RUN SPECTRAL MEASUREMENTS MODULE VERSION 2.1

DIM Run$(200)

4044 IF NPAR<2 THEN
4046 Run$="Spectral attenuation measurements in process"
4048 ELSE
4050 Run$=Runmsg$
4052 END IF
4054 Spot_flag=1
4056 !Check for over-filled launch specification
4058 IF NPAR>0 THEN
4060 IF (POS(Spot$,"F")<>0 OR POS(Spot$,"f")<>0) THEN
4062 Spot_flag=0
4064 Run$=Run$&"Launch overfilled."
4066 END IF
4068 CALL Rundisplay(Run$)
4070 CALL Specmeas(0,Spot_flag)
4072 CALL Cleardisplay
4074 SUBEND

SUB Spectref(OPTIONAL Direct$,Runmsg$)

SPECTRAL ATTENUATION REFERENCE MEASUREMENTS MODULE VERSION 2.1

DIM Run$(200)
4092 Crlf$=CHR$(13)&CHR$(10)
4094 Spot_flag=1
4096 IF NPAR>0 THEN
4098 IF (POS(Direct$,"F")<>0 OR POS(Direct$,"f")<>0) THEN Spot_flag=0
4100 IF Direct$ was included, so check for a "D" or "d"
4102 IF POS(Direct$,"D") OR POS(Direct$,"d") THEN
4104 IF POS(Direct$,"U") OR POS(Direct$,"u") THEN
4106 A Direct Uncorrected run is requested.
4108 IF NPAR<2 THEN
4110 Run$="Collecting uncorrected direct reference data"&Crlf$&"for spectral attenuation."
4112 ELSE
4114 Run$="Collecting uncorrected direct reference data"&Crlf$&Runmsg$
4116 END IF
4118 IF Spot_flag=0 THEN Run$=Run$&"Launch overfilled."
4120 CALL Rundisplay(Run$)
4122 CALL Specmeas(2,Spot_flag)
4124 CALL Cleardisplay
4126 ELSE
4128 Run$="Collecting uncorrected direct reference data"&Crlf$&Runmsg$
4130 END IF
4132 IF Spot_flag=0 THEN Run$=Run$&Crlf$&"Launch overfilled."
4134 CALL Rundisplay(Run$)
4136 CALL Specmeas(2,Spot_flag)
4138 CALL Cleardisplay
4140 ELSE
4142 ! Uncorrected NOT specified, do a corrected direct reference run.
4144 !
4146 IF NPAR<2 THEN
4148 Run$="Collecting direct reference data for spectral attenuation."
4150 CALL Rundisplay(Run$)
4152 CALL Specmeas(0,Spot_flag)
4154 CALL Cleardisplay
4156 ELSE
4158 Run$="Collecting direct reference data for spectral attenuation."
4160 CALL Rundisplay(Run$)
4162 CALL Specmeas(0,Spot_flag)
4164 CALL Cleardisplay
4166 ELSE
4168 ! Uncorrected NOT specified, do a corrected direct reference run.
4170 !
END IF
IF Spot_flag=0 THEN Run$=Run$&Crlf$&"Launch overfilled."
CALL Rundisplay(Run$)
CALL Specmeas(3,Spot_flag)
CALL Cleardisplay
END IF
ELSE
GOTO Ref
END IF
ELSE
! The Direct$ string did not contain a "D" or "d" or the direct$ parameter was not specified, so do a short-fiber reference run.
Ref:
IF NPAC2 THEN
RUN$="Collecting spectral attenuation"&Crlf$&"reference data."
ELSE
RUN$="Collecting &Runmsg$" reference data."
END IF
IF Spot_flag=0 THEN Run$=Run$&Crlf$&"Launch overfilled."
CALL Rundisplay(Run$)
CALL Specmeas(1,Spot_flag)
CALL Cleardisplay
END IF
SUBEND
SUB Specmeas(Run_flag,OPTIONAL Spot)
*******************************************************************************
SPECTRAL ATTENUATION MEASUREMENTS MODULE VERSION 2.1P
*******************************************************************************
COM /Diskdrive/ Sysdrive$,Arcdrive$
COM /Sysdate/ Serial_num$,Laser_flag,Filter_flag,Filter(*),Num_focus Focus(*),Cutoff,Low_wave,High_wave,Det_switch
COM /Fiber/ Fiber_id$,Fiber_len,Log_time$
COM /Wavelength/ Wavelength(*),Numsteps
COM /Specrundate/ Specrundata(*),Specrun_id$
COM /Specrdata/ Specrdata(*),Specref_id$
COM /Directref/ Specrefcor(*),Pulse_refcor(*),Pulse_corwave(*),Correct_flag(*)
COM /Cutoff/ Cutref(*),Cutresult(*),Cutoff_id$,Cutoff_wave,First,Last,Signal,Intercept
DIM Filename$(25)
INTEGER Index
Set-up instruments for Spectral Measurements
IF NPAR=2 THEN
Spot_flag=Spot
ELSE
Spot_flag=1
END IF
IF Filter_flag<>1 THEN
CALL F2000send("WAV COUPL")
ELSE IF Filter_flag=1 means bandpass filters used and not monochromator.
CALL F2000send("FIL COUPL")
END IF
IF Spot_flag THEN
CALL F2000send("SPOT-IN")
ELSE
CALL F2000send("SPOT-OUT")
END IF
ELSE CALL F2000Send("XMIT")
END IF
CALL F2000Send("VOUT TARGET-OUT FF-OUT",1)
SELECT Run_flag

| For fiber measurements, store data in Specrundata array. |
| For reference or direct measurements, store data in Specrefdata array. |

CASE 0
    Specrun_id=Fiber_id"&Log_time$
    Specrundata(0,0)=Numsteps
    Specrundata(0,1)=Fiber_len
END CASE

CASE 1,2,3
    Specref_id=Fiber_id"&Log_time$
    Specrefdata(0,0)=Numsteps
    Specrefdata(0,1)=Fiber_len
END CASE

CASE 4
    Cutoff_id=Fiber_id"&Log_time$
END CASE

END SELECT

Run measurements at each wavelength in the wavelength array

FOR Index=0 TO Numsteps-1
    Make measurement at each wavelength
    Set the FOA-2000 to the next wavelength
    CALL Nextwave(Wavelength(Index))
    IF Run_flag=0 OR Run_flag=1 OR Run_flag=4 THEN
        CALL Setfocus(Wavelength(Index))
    END IF

Make the measurement at this wavelength

Measurement=FNVolimeter(.01)

Now store the measurement in the appropriate common array

SELECT Run_flag
    Specrundata(Index+1,0)=Wavelength(Index)
    Specrundata(Index+1,1)=Measurement
END CASE

CASE 1,2,3
    Specrefdata(Index+1,0)=Wavelength(Index)
    Specrefdata(Index+1,1)=Measurement
END CASE

CASE 4
    Cutref(Index+1,0)=Wavelength(Index)
    Cutref(Index+1,1)=Measurement
END CASE

END SELECT

NEXT Index

End of measurement loop

CALL F2000Send("0 IN-X 0 IN-Y 0")

For corrected direct measurements, the direct data must be multiplied by the launch correction factors stored in the common array Specrefcor.

IF Run_flag=3 THEN
4412 Dead2: GOTO Dead2
4414 ELSE
4416 Ioffset=0
4418 FOR Index=1 TO Numsteps
4420 WHILE Specrefdata(Index,0)<Specrefcor(Index+Ioffset,0)
4422 Ioffset=Ioffset+1
4424 IF Index+Ioffset>Specrefcor(0,0) THEN
4426 BEEP
4428 PRINT TABXY(1,17);"SPECMEAS -- A correction factor was not found for a wavelength used in"
4430 PRINT TABXY(1,18);"the direct-spot measurements."
4432 Dead1: GOTO Dead1
4434 ENIF
4436 EN WHILE
4438 Specrefdata(Index,1)=Specrefdata(Index,1)*Specrefcor(Index+Ioffset,1)
4440 NEXT Index
4442 END IF
4444 END IF
4446 SUBEND
4448 !
4450 !
4452 SUB Speccor
4454 !+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
4456 ! CALCULATE SPEC ATTEN DIRECT CORRECTION FACTORS MODULE  VERSION 2.1P
4458 !-----------------------------------------------------------------------------------------------
4460 COM /Diskdrive/ Sysdrive$ ~cdrive$
4462 COM /Specrundata/ Specrun..a(*),Specrun.id$
4464 COM /Specrefdata/ Specrefdata(*),Specref.id$
4466 COM /Directref/ Specrefcor(*),Pulserefcor(*),Pulsecorwave(*),Correct_flag(*)
4468 Filename$="speccor"
4470 INTEGER Index
4472 Specrefcor(0,0)=Specrefdata(0,0)
4474 Specrefcor(0,1)=Specrefdata(0,1)
4476 FOR Index=1 TO Specrefdata(0,0)
4478 IF Specrefdata(Index,0)>Specrefdata(Index,0) THEN
4480 BEEP
4482 DISP "SPECCOR -- Short fiber and direct data wavelengths do not match."
4484 Dead1: GOTO Dead1
4486 ELSE
4488 Specrefcor(Index,0)=Specrefdata(Index,0)
4490 Specrefcor(Index,1)=Specrefdata(Index,1)/Specrefdata(Index,1)
4492 END IF
4494 NEXT Index
4496 !
4498 ! Write the new data in the file called "speccor"
4500 !
4502 ON ERROR GOSUB File_err
4504 CREATE BDAT Filename$%Sysdrive$,210,8
4506 ASSIGN @Outfile TO Filename$%Sysdrive$
4508 OFF ERROR
4510 OUTPUT @Outfile;Specrefcor(*)
4512 ASSIGN @Outfile TO *
4514 GOTO Done
4516 File_err:IF ERRN=54 THEN
4518 PURGE Filename$%Sysdrive$
4520 ELSE
4522 PRINT "SPECCOR -- Err. r number "&VAL$(ERRN)
4524 Dead2:GOTO Dead2
4526 END IF
SUB Specatcomp
******************************************************************************
SPECTRAL ATTENUATION COMPUTE MODULE                                  VERSION 2.1
******************************************************************************
COM /Spacrunidata/ Specrunidata(*),Specrun_id$
COM /Specrefdata/ Specrefdata(*),Specref_id$
COM /Specattdata/ Specattdata(*),Specatt_id$
INTEGER Index
CALL Rundisplay("Computing Spectral Attenuation Results.")
Length=Specrundata(0,1)  ! If fiber length is not given,
IF Length=0 THEN Length=4.6  ! then use 1 for length.
Specattdata(0,0)=Specrundata(0,0)  'Store the number of points
Specattdata(0,1)=Specrundata(0,1)  'Store the fiber length
Specatt_id$=Specrun_id$  'Store the fiber id string

Now compute the results at each wavelength
FOR Index=1 TO Specrundata(0,0)
  ! Find the wavelength in the REF sample that corresponds to the RUN.
  Index1=1
  WHILE Specrundata(Index,0)<Specrefdata(Index,0) AND Index1<Specrefdata(0,0)
    Index1=Index1+1
  END WHILE
  IF Index1>Specrefdata(0,0) THEN
    BEEP
    PRINT TABXY(17,1);"SPECATCOMP -- The reference does not contain a wavelength found in the measurement. Program idle."
  END IF
  Specattdata(Index,0)=Specrundata(Index,0)  'Record the wavelength
  IF Specrefdata(Index,1)/Specrundata(Index,1)<0 THEN
    Specattdata(Index,1)=-100
  ELSE
    Specattdata(Index,1)=10*LOG(Specrefdata(Index,1)/Specrundata(Index,1))
  END IF
  END IF
  Specattdata(Index,1)=Specattdata(Index,1)/Length
NEXT Index
CALL Cleardisplay
SUBEND
******************************************************************************
SPECTRAL ATTENUATION OUTPUT LISTING MODULE                                    VERSION 2.1
******************************************************************************
DIM Title*(251
INTEGER I
REAL Divby
COM /Ipasts/ @Foa2000@Egg5205@Tek496p@Tek7954,Printer_add
COM /Specattdata/ Specattdata(*),Specatt_id$

Now set-up the table output
OUTPUT KBD USING ",",""  ! Set up screen for the table

IF NPAR>0 THEN

IF POS(Print_flag$,"MET") THEN Divby=1000

IF POS(Print_flag$,"TEN") THEN Divby=100

IF POS(Print_flag$,"HUN") THEN Divby=10

IF POS(Print_flag$,"KIL") THEN Divby=1

IF POS(Print_flag$,"P") OR POS(Print_flag$,"p") THEN Print_it

END IF

GOSUB Print_tbl

ON KEY 8 LABEL "PRINT" GOTO Print_it ! HARDCopy?

ON KEY 5 LABEL "CONTINUE" GOTO Done

BEEP

Wait_here: GOTO Wait_here

Print_it: OFF KEY

PRINTER IS Printer_add

GOSUB Print_tbl

PRINT ! Put some white space at the bottom

PRINTER IS !

GOTO Done

Print_tbl: !

IF NPAR<2 THEN

PRINT "SPECTRAL ATTENUATION!"

PRINT "------------------"

ELSE

PRINT Newtitle$

END IF

PRINT "FIBER ID: "&Specatt_id$

PRINT "LENGTH: "&Specattdata(0,1)" km"

PRINT "WAVELENGTH "&Title$

PRINT

FOR I=1 TO Specattdata(0,0)

PRINT USING "4.4D,10X,6.4D"Specattdata(I,0),Specattdata(I,1)/Divby

NEXT I

RETURN

Done: OUTPUT KBD USING ",K"","K"

Clear the screen

SUBEND

SUB Specatplot(OPTIONAL Print_flag$, Lowwave, Highwave, Newtitle$)

!--------------------------------------------

COM /Iopaths/ @Foxx2000, @Egg5205, @Tek496p, @Tek7854, Printer_add

COM /Specattdata/ Specattdata(*), Specatt_id$

INTEGER Index

INTEGER Lowave

REAL Divby

REAL Divby

DIM Xlabel$(40), Ylabel$(40)

Divby=1

! In case this parameter isn't passed
4794 IF POS(Print_flag$,"KILO") THEN Divby=1
4795 END IF
4798 !
4800 !Initialize plotting labels and limits
4802 !
4804 Xlabel$="Wavelength (um)"
4806 IF Specattdata(0,1)=0 THEN
4808 Ylabel$="dB"
4810 ELSE
4812 IF Divby=1 THEN Ylabel$="dB/km"
4814 IF Divby=10 THEN Ylabel$="dB/10m"
4816 IF Divby=100 THEN Ylabel$="dB/100m"
4818 IF Divby=1000 THEN Ylabel$="dB/m"
4820 END IF
4822 !
4824 IF (NPAR>1) THEN
4826 IF Low_wave>O THEN
4828 Minx=Low_wave
4830 ELSE
4832 Minx=Specattdata(0,0)
4834 END IF
4836 ELSE
4839 IF Print_flag$="2" THEN
4840 INPUT "Enter the minimum wavelength value in nm: ",Minx
4842 ELSE
4844 Minx=Specattdata(0,0)
4846 END IF
4848 END IF
4850 IF (NPAR>2) THEN
4852 IF High_wave>0 THEN
4854 Maxx=High_wave
4856 ELSE
4858 Maxx=Specattdata(0,0)
4860 END IF
4862 ELSE
4864 IF Print_flag$="2" THEN
4866 INPUT "Enter the MAXIMUM wavelength value in nm: ",Maxx
4868 ELSE
4870 Maxx=Specattdata(0,0)
4872 END IF
4874 END IF
4876 Minx=INT(Minx/100)*100
4878 Maxx=INT((Maxx+99)/100)*100
4880 Miny=0
4882 Maxy=0
4884 FOR Index=1 TO Specattdata(0,0)
4886 IF (Specattdata(Index,1)/Divby)>Maxy THEN Maxy=Specattdata(Index,1)/Divby
4888 NEXT Index
4889 !
4891 Minx=INT(Minx/5)*5
4893 IF Maxy<5 THEN Maxy=5
4894 IF Maxy>20 THEN Maxy=20
4896 Ticx=(Specattdata(0,0)-Specattdata(1,0))/10
4898 Ticy=INT(Ticx/10)*10
4900 Ticy=1
4902 !
4904 !Initialize screen, set line type to dotted, and draw the grid
4906 !
4908 GINIT
4910 GCLEAR
4912 GRAPHICS ON
4914 CSIZE 5,.55
4924 IF Specattdata(0,1)=0 THEN
4926 LABEL "SPECTRAL ATTENUATION"
4928 ELSE
4930 LABEL USING ""SPECTRAL ATTENUATION LENGTH: "",DD.DDDD,"" $m$";
Specattdata(0,1)
4932 END IF
4934 ELSE
4936 LABEL "&Newtitle$
4938 END IF
4940 PRINT TABXY(1,2);
4942 PRINT "ID: &Specatt_id$;
4944 PRINT USING "$2X,8A,DQ.DDD,3A";"Length: ",Specattdata(0,1)," $m$"
4946 VIEWPORT 20,92*RATIO,30,86
4948 WINDOW Minx,Maxx,Miny,Maxy
4950 LINE TYPE 4
4952 GRID Ticx,Ticy,Minx,Miny
4954 |
4956 |Reset the line type to solid, and plot the data
4958 |
4960 LINE TYPE 1
4962 FOR Index=1 TO Specattdata(0,0)
4964 PLOT Specattdata(Index,0),Specattdata(Index,1)/Divby
4966 NEXT Index
4968 |
4970 |Set the label mode to center, units to degrees, rotation to zero
4972 |degrees, and expand the hard clip to make room for the labels.
4974 |Then set the label size for the x-axis.
4976 |
4979 LORG 5
4980 DEG
4982 LDIR 0
4984 VIEWPORT 0,100*RATIO,0,100
4986 WINDOW 0,100*RATIO,0,100
4988 CSIZE 6,.6
4990 |
4992 |Label the x-axis
4994 |
4996 MOVE 70,18
4998 LABEL Xlabel$
5000 CSIZE 4,.65
5002 LORG 6
5004 FOR Xpos=20 TO 123 STEP 100*Ticx/(Maxx-Minx)*2
5006 MOVE Xpos,29
5008 LABEL USING "$O.DO$;$(Xpos-20)/100*(Maxx-Minx)/1000+Miny/1000
5010 NEXT Xpos
5012 |
5014 |Change to the Y-axis, put the title on the y-axis, then
5016 |label the grid marks on the y-axis.
5018 |
5020 CSIZE 6,.6
5022 LORG 5
5024 MOVE 8.53+LEN(Ylabel$)*3
5026 FOR I=1 TO LEN(Ylabel$)
5028 LABEL Ylabel$[I+1]
5030 NEXT I
5032 CSIZE 4,.6
5034 LORG 8
5036 FOR Ypos=30 TO 88 STEP 56*Ticy/(Maxy-Miny)
5038 MOVE 19,Ypos
5040 LABEL USING "$O.DO$;$(Ypos-30)/56*(Maxy-Miny)+Miny
5042 NEXT Ypos
5044 |
...
PROGRAM NEAR-FIELD MEASUREMENTS RUN MODULE

DIMENSION Nearfield(0), Nearfieldval(0), Num_points, Setting, Index, Maxloc

CALL Rundisplay("Near-field measurements in progress.")

CALL F2000send("GERMAIN LED ON CHOP-ON SPOT-OUT XMIT PIN-IN")
CALL F2000send("OUT @ ATTENUAT FF-OUT")
CALL F2000send("OUT COUPL")

CALL F2000send("VAL(Pin_y)& " OUT-Y "&VAL$(Pin_z)& "OUT-Z")

CALL Setscale(.1, Peakval)

CALL F2000send("0 OUT-X 0 OUT-Y 0 OUT-Z")

INDEX Index, Setting, Meas_range, Maxloc

INPUT KB USING ":,K": "K"

IF (Nearfield(Index,1)) > Maxval THEN
  Maxval = Nearfield(Index,1)
  Maxloc = Index
END IF

Now put the stage back to the original zero value before correcting for the pinhole position.

CALL F2000send("0 OUT-X 0 OUT-Y 0 OUT-Z")
SUB Nearfieldplot(OPTIONAL Print_flag$)

"NEAR FIELD GRAPHICS OUTPUT MODULE
VERSION 2.1"

INTEGER Index
COM /Nearfield/ Nfieldval(*),Num_points,Nearfield(*),Nfield_id$
DIM Title$(40),Xlabel$(40)

SUBEND

CALL Core_diam(Nearfield(*),Core_diam)
Core_diam$=INT(10*Core_diam+.5) !Round the results
Title$="NEAR FIELD PATTERN"
Xlabel$="Distance (um)"
GINIT
GCLEAR
GRAPHICS ON
VIEWPORT 0,1000,RATIO,10,100
MOVE 0,95.5
CSIZE 5
LABEL Title$
LABEL "ID: "&Nfield_id$
Minx=100
Maxx=100
Miny=0
Maxy=1
Xsize=ABS(Maxx-Minx)
Ysize=ABS(Maxy-Miny)
Botborder=Miny-.2*Ysize ! Create a graph layout with space for
Topborder=Maxy+.1*Ysize ! labels
Leftborder=Minx-.2*Xsize
Rgtborder=Maxx+.05*Xsize
VIEWPORT 0,1000,RATIO,18,95
WINDOW Leftborder,Rgtborder,Botborder,Topborder

** Generate the frame **
MOVE Minx,Miny
IDRAW Xsize,0
IDRAW 0,Ysize
IMOVE -Xsize,0
IDRAW 0,-Ysize

** Generate the graticule lines **
LINE TYPE 4 !Graticule in dotted lines
FOR Index=0 TO 5
  MOVE Minx,Miny+(Ysize*Index/5)
  IDRAW Xsize,0
  IDRAW 0,-Ysize
  IMOVE -Xsize,0
NEXT Index
FOR Index=0 TO 8
  MOVE Minx+(Xsize*Index/8),Miny
  IDRAW Ysize,0
  IDRAW 0,-Ysize
NEXT Index
LINE TYPE 1 !Back to solid lines
MOVE Nearfield(1,0),Nearfield(1,1)
FOR Index=2 TO Nearfield(0,0)
DRAW Nearfield(Index,0),Nearfield(Index,1)
NEXT Index

** Put in the X-axis graticule labels **

CSIZE 4
FOR Index=0 TO 8
Value=Minx+Index*Xsize/8
MOVE Value-.06*Xsize,Miny-.4*(Miny-Botborder)
LABEL USING "S3D":"Value
NEXT Index

Xpos=Minx+Xsize/2-LEN(Xlabel$)*(Xsize/40)/2 !Compute place for XLABEL$

** Print the X label string **

MOVE Xpos,Botborder
CSIZE 5
LABEL Xlabel$

** Print the core diameter **

WINDOW 0,100*RATIO,10,100
VIEWPORT 0,100*RATIO,10,100
MOVE 0,9
IF Core_diam=0 THEN
LABEL "Core diameter not found"
ELSE
LABEL USING """Core diameter = """,DDD.DD"";Core_diam
END IF

IF NPAR=1 THEN
IF UPCS(Print_flag$[1,1])="P" THEN Print_it
END IF
ON KEY 8 LABEL "PRINT" GOTO Print_it
ON KEY 5 LABEL "CONTINUE" GOTO Done
Wait_here:GOTO Wait_here
Print_it:OFF KEY
OUTPUT KBD USING ",,K";"
DUMP GRAPHICS
OUTPUT KBD USING ",,K";"
Done:GCLEAR
GRAPHICS OFF
SUBEND

SUB Core_diam(Nearfield(*),Diameter)
COMPUTE CORE DIAMETER VERSION 2.1
This module computes core diameter on the near-field pattern.
Threshold=.025 ! Use 2.5% points

First, locate the 15% points to be sure we are off the noise floor.
Diameter=0
Index=1
WHILE Nearfield(Index,1)<.15
Index=Index+1
IF Index>Nearfield(0,0) THEN Done
END WHILE
WHILE Nearfield(Index,0)>Threshold
    Index=Index+1
    IF Index>Num_points THEN Done
END WHILE

FOR Index=0 TO Num_points-1
    IF Ffieldval(Index)<-550*Farfield_step THEN
        BEEP
        PRINT TABXY(1,17);"FFIELDVALS -- A value less than ";Lower_limit;" was specified."
    END IF
    IF Ffieldval(Index)>140*Farfield_step THEN
        BEEP
        PRINT TABXY(1,17);"FFIELDVALS -- A value greater than ";Upper_limit;" will be ignored.
    END IF
    FOR Index=Index-1 TO 0
        IF Ffieldval(Index)-Upper_limit THEN
            BEEP
            PRINT TABXY(1,17);"FFIELDVALS -- A value greater than ";Upper_limit;" was specified.";
        ELSEIF Ffieldval(Index)<Lower_limit THEN
            BEEP
            PRINT TABXY(1,17);"FFIELDVALS -- A value less than ";Lower_limit;" was specified."
            Ffieldval(Index)=-550*Farfield_step
        END IF
    END FOR
NEXT Index
WAIT 3
Num_points=200
FOR Index=0 TO Num_points-1
    IF Ffieldval(Index)<-550*Farfield_step THEN
        BEEP
        PRINT TABXY(1,17);"FFIELDVALS -- A value less than ";Lower_limit;" was specified.";
    END IF
    IF Ffieldval(Index)>140*Farfield_step THEN
        BEEP
        PRINT TABXY(1,17);"FFIELDVALS -- A value greater than ";Upper_limit;" will be ignored.
    END IF
NEXT Index
5662     WAIT 3
5664     Fieldval(Index)=140*Farfield_step
5666     END IF
5668     NEXT Index
5700     DISP " "
5702     SUBEND
5704     !
5706     !
5708     SUB Fieldrun(Ffwave,OPTIONAL Runflag$)
5710     !+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
5712     ! FAR-FIELD MEASUREMENTS RUN MODULE  VERSION 2.1
5714     !+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
5716     COM /Farfield/ Fieldval(*),Num_points,Farfield(*),Field_id$
5718     COM /Fiber/ Fiber_id$,Fiber_len,Log_time$
5720     COM /Syscal/ Pin_x,Pin_y,Pin_z,Inx_step,Iny_step,Outx_step,Outy_step,Farfield_step,Lfnoise
5722     COM /Farfield_wave/ Ffwave
5724     INTEGER Index,Setting,Meas_range,Runflag
5726     REAL Sintheta,Delta
5728     Ffwave=Ffwave
5730     IF NPAR<2 THEN
5732     Runflag=1  ! defaults to using scanner edge if not specified
5734     ELSE
5736     IF POS(Runflag$,"PIN") THEN
5738     Runflag=0
5740     ELSE
5742     Runflag=1
5744     END IF
5746     END IF
5748     OUTPUT KBD USING ".K","K"
5750     IF Runflag THEN
5752     CALL Rundisplay("Far-field measurements in progress.
          (Using scanner edge.)")
5754     ELSE
5756     CALL Rundisplay("Far-field measurements in progress.
          (Using pinhole.)")
5758     END IF
5760     !
5762     ! Set up instruments for far field measurements
5764     !
5766     CALL Nextwave(Ffwave)
5768     CALL F2000send("INSB LAMP LAMP-ON CHOP-ON SPOT-OUT XMIT FF-IN")
5770     CALL F2000send("VOUT TARGET-OUT 0 ATTENUAT",1)
5772     CALL F2000send("FF COUPL",1)
5774     CALL F2000send("-550 FAR-FIELD",1) ! eliminate backlash
5776     !
5778     ! Measure the approx peak amplitude and fix the lock-in voltmeter scale.
5780     !
5782     Reading=FNVoltmeter(.1)
5784     Peakval=2*Reading
5786     CALL Setscale(.1,Peachval)
5788     !
5790     ! Take the measurements at each specified far-field position
5792     !
5794     Delta=1.59 ! Positional correction of edge of Far Field Scanner
5796     ! (Actually this $ is delta/focal_length. delta=0.4281")
5798     ! The value of Delta will affect how well the center of
5800     ! the far field plot lines up with the peak intensity of
5802     ! the output intensity pattern.
5804     CALL Rundisplay("
Scanning edge technique

Setting=(Sintheta-Delta)/Farfield_step
Actual=(Setting*Farfield_step+Delta)
CALL F2000send(VAL$(Setting)" FAR-FIELD",1)
Reading=FNVolmeter(.05)
Farfield(Index,1)=Reading
ELSE
Pinhole technique
Corrects measurement for COS(PHI)
Setting=Sintheta/Farfield_step
Actual=Setting*Farfield_step note integer truncation
CALL F2000send(VAL$(Setting)" FAR-FIELD",1)
Reading=FNVolmeter(.05)
Farfield(Index,1)=Reading
ENDIF

PRINT Actual,Reading
The following factor of .873 is a calibration factor. It was derived by comparing a numerical aperture measurement made on this machine with one made on George McCabe's NA measurement station. The fiber used for comparison was 900228 on 28 Mar 90.

Farfield(Index,0)=Actual*.873
NEXT Index

Ffield_id$=Fiber_id$" Log_time$ Store the fiber ID & time
Farfield(0,0)=Num_points Also store # points here (for ffieldplot)
CALL F2000send("-550 FAR-FIELD") Move scanner back down
CALL Cleardisplay

SUBEND

SUB Ffieldplot(OPTIONAL Print_flag$,New_title$)

OUTPUT GRAPHICS MODULE FOR FAR-FIELD PATTERN VERSION 2.1
This module is responsible for both plotting and printing all Far Field data, including raw data, differentiated data, and smoothed data.

COM /Farfield/ Ffieldval(*),Num_points,Farfield(*),Ffield_id$
COM /Fftempdata/ Ffrawdata(*),FfDiffdata(*),Ffsmothdata(*)
COM /Farfield_wave/ Ffwavelan
INTEGER I,J,Index
DIM Title$(80),Xlabel$(40)

Create a file in which to store the raw data. This is the file which is loaded back into the Farfield(*) array if the store option is chosen.
Also create a file for the differentiated, rough (not smoothed) data.
Either the raw or rough data can be smoothed.

IF Print_flag$="RAW DATA" THEN
FOR I=0 TO Farfield(0,0)
FOR J=0 TO I
Ffrawdata(I,J)=Farfield(I,J)
NEXT J
NEXT I
END IF

Compute the NA for differentiated or smoothed, differentiated data only.

IF Print_flag$="DIFF" THEN
CALL Normalize("DIFF",Num_points)
CALL Numaper("SMOOTH",Num_aper)  
Num_aper=.001*INT(1000*Num_aper+.5)  
Round to 3 places

END IF

Now plot the data.

Plotit:

Xlabel="SIN(angle)"
Title="Far Field Pattern"
IF NPAR>1 THEN Title$=New_title$
INIT
GCLEAR
GRAPHICS ON
VIEWPORT 0,100*RATIO,10,100
MOVE 0,95.5
CSIZE 5
LABEL Title$
LABEL * ID: &Field_id$
Minx=-.3
Maxx=.3
Miny=0
Maxy=1
Xsize=ABS(Maxx-Minx)
Ysize=ABS(Maxy-Miny)
Botborder=Miny-.2*Ysize  
Topborder=Maxy+.1*Ysize  
Leftborder=Minx-.2*Xsize
Rgtborder=Maxx+.05*Xsize
VIEWPORT 0,100*RATIO,20,95
WINDOW Leftborder,Rgtborder,Topborder,Botborder

** Generate the frame **

MOVE Minx,Miny
IDRAW Xsize,0
IDRAW 0,Ysize
MOVE -Xsize,0
IDRAW 0,-Ysize

** Generate the graticule lines **

LINE TYPE 4
FOR Index=0 TO 5
MOVE Minx,Miny+(Ysize*Index/5)
IDRAW Xsize,0
NEXT Index
FOR Index=0 TO 6
MOVE Minx+(Xsize*Index/6),Miny
IDRAW 0,Ysize
NEXT Index
LINE TYPE 1

** Draw the graph itself

IF Print$="RAW DATA" THEN
MOVE Ffrawdata(0,0),Ffrawdata(1,1)
FOR Index=2 TO Ffrawdata(0,0)
DRAW Ffrawdata(Index,0),Ffrawdata(Index,1)
NEXT Index
END IF

IF Print$="DIFF" THEN


END IF

IF Print_flag$="SMOOTH" THEN

MOVE Ffsmoothdata(1,0),Ffsmoothdata(1,1)

FOR Index=2 TO Ffsmoothdata(0,0)

DRAW Ffsmoothdata(Index,0),Ffsmoothdata(Index,1)

NEXT Index

END IF

** Put in the X-axis graticule labels **

CSIZE 4

FOR Index=0 TO 6

Value=Minx+Index*Xsize/6  !Compute the value of the label

MOVE Value-.09*Xsize,Miny-.4*(Miny-Botborder)

LABEL USING "2D.2D";Value

NEXT Index

Xpos=Minx*Xsize/2-LEN(Xlabel$)*(Xsize/40)/2  !Compute place for XLABEL$

** Print the X label string **

CSIZE 5

MOVE Xpos,Botborder

LABEL Xlabel$

** Print numerical aperture value **

WINDOW 0,100*RATIO,10,100

VIEWPORT 0,100*RATIO,10,100

MOVE 0,12

IF Print_flag$="DIFF" OR (Print_flag$="SMOOTH" AND Ffdiffdata(0,0)=0) THEN

IF Num_aper=0 THEN

LABEL " Numerical Aperture Not Found."

ELSE

IF Ffwavelen=0 THEN

LABEL USING " 7X,4A,D.DDD,3X,13A,DDDDD,3A";"NA =",Num_aper

ELSE


END IF

END IF

END IF

! The rest of this is concerned with where to go after the screen plot.

Key_guys: !

Keys which appear on every plot:

ON KEY 1 LABEL " SMOOTH DATA" GOTO Smooth

ON KEY 2 LABEL "DIFFERENTIATE" GOTO Done

ON KEY 4 LABEL " STORE RAW DATA" GOTO Storeit

ON KEY 5 LABEL "EXIT" GOTO Exit_all

ON KEY 7 LABEL "PRINT LISTING" GOTO Print_list

ON KEY 8 LABEL "PRINT PLOT" GOTO Print_plot

Wait_key_guys:GOTO Wait_key_guys

Smooth: OFF KEY

PRINT TABXY(15,10);"Smooth the RAW, DIFFerentiated or SMOOTHED data?"

ON KEY 1 LABEL "RAW" GOTO Raw_smooth

ON KEY 2 LABEL "SMOOTH" GOTO Smooth

ON KEY 3 LABEL "DIFFERENTIATE" GOTO Diff

ON KEY 4 LABEL "STORE RAW DATA" GOTO Storeit

ON KEY 5 LABEL "EXIT" GOTO Exit_all

ON KEY 6 LABEL "PRINT LISTING" GOTO Print_list

ON KEY 7 LABEL "PRINT PLOT" GOTO Print_plot

Wait_key_guys:GOTO Wait_key_guys
6204 Raw_smooth: OFF KEY
6206 CALL Ffsmooth("RAW DATA")
6209 Print_flag$="SMOOTH"
6210 GOTO Plotit
6212 Diff_smooth: OFF KEY !If we haven't computed diff data, then do that first
6214 IF Ffdiffdata(0,0)=0 THEN
6216 PRINT USING "6X,70A";"Differentiated data has not been calculated. Can't be smoothed yet."
6218 GOTO Smooth
6220 END IF
6222 CALL Ffsmooth("DIFF")
6226 PRINT USING "6X,70A";"Smoothed data has not been calculated. Can't be smoothed yet."
6228 GOTO Smooth
6230 Smooth_smooth: OFF KEY
6232 IF Ffsmoothdata(0,0)=0 THEN
6234 PRINT USING "6X,70A";"Smoothed data has not been calculated. Can't be smoothed yet."
6236 GOTO Smooth
6240 END IF
6242 CALL Ffsmooth("SMOOTH")
6244 Print_flag$="SMOOTH"
6246 SUBEXIT
6250 !------------------------------------------------------------------
6252 !
6254 Storeit: OFF KEY
6256 CALL Cleardisplay
6258 CALL Archive
6260 GOTO Plotit
6262 !------------------------------------------------------------------
6266 !
6268 Print_list: OFF KEY
6270 PRINT TABXY(15,10);"Smooth the RAW, DIFFerentiated or SMOOTHED data?"
6272 ON KEY 1 LABEL " RAW" GOTO Print_raw
6274 ON KEY 4 LABEL " DIFF" GOTO Print_diff
6276 ON KEY 8 LABEL "SMOOTHED" GOTO Print_smooth
6278 Nogo: GOTO Nogo
6280 !
6282 Print_raw: OFF KEY
6284 PRINT IS PRT
6286 PRINT " Far Field Raw Data for Fiber:",Ffield_id$
6288 PRINT "";
6290 PRINT USING "15X,48A";"Number, Scanner Position, Normalized Signal"
6292 PRINT " 
6294 FOR I=1 TO Ffrawdata(0,0)
6296 PRINT USING "16X,DDD,10X,M.ddd,15X,M.D.3D";I,Ffrawdata(1,I),Ffrawdata(1,1)
6298 NEXT I
6300 PRINT ""
6302 PRINTER IS CRT
6304 CALL Cleardisplay
6306 GOTO Plotit
6308 !
6310 Print_diff: OFF KEY
6312 IF Ffdiffdata(0,0)=0 THEN
6314 PRINT USING "6X,70A";"Differentiated data has not been calculated. Can't be printed yet."
6316 GOTO Print_list
6318 GOTO Nogo
6320 END IF
6322 !PRINTER IS PRT
6324 !
6326 !
6328 !
6330 !
6332 !
6334 !
6336 !
6338 !
6340 !
6342 !
6344 !
6346 !
6348 !
6350 !
6352 !
6354 !
6356 !
6358 !
6360 !
6362 !
6364 !
6366 !
6368 !
6370 !
6372 !
6374 !
6376 !
6378 !
6380 !
6382 !
6384 !
6386 !
6388 !
6390 !
6392 !
6394 FOR I=1 TO F...
FOR I=1 TO Ffdiffdata(0,0)
    PRINT USING "15X,10X,M.3D,15X,M.3D";I,Ffdiffdata(I,0),Ffdiffdata(I,1)
NEXT I
PRINT " "
PRINTER IS CRT
CALL Cleardisplay
GOTO Plotit

PRINT smooth: OFF KEY
IF Ffsmoothdata(0,0)=0 THEN
    PRINT USING "6X,70A";"Smoothed data has not been calculated. Can't be printed yet."
GOTO Print_list
END IF
PRINTER IS PRT
PRINT Far Field Smoothed Data for Fiber:,"Ffield_id$
PRINT "Far Field Smoothed Data for Fiber:",Ffield_id$
PRINT "Number Scanner Position Normalized Signal"
PRINT " "
FOR I=1 TO Ffsmoothdata(0,0)
    PRINT USING "15X,48A";I,Ffsmoothdata(I,e),Ffsmoothdata(I,1)
NEXT I
PRINT " 
PRINTER IS CRT
CALL Cleardisplay
GOTO Plotit

PRINT_plot: OFF KEY
OUTPUT KBD USING ";,K";"
DUMP GRAPHICS
OUTPUT KBD USING ";,K";"
GOTO Plotit

Exit_all: OFF KEY
CALL Menu

Done: OFF KEY
IF Print_flag$="DIFF" THEN
    PRINT " 
    PRINT "Sorry to deceive you, you can't differentiate this data."
GOTO Plotit
END IF
GOTO Plotit

SUB EnAperture(Print_flag$,Num_aper)

----------------------------------------
This module calculates the fiber NA using a variable threshold method
(presently set at 5%). The input data set may either be the rough
differentiated data, or a smoothed version of the same.
----------------------------------------

NUMAPER (Print_flag$, Num_aper)

----------------------------------------
Compute Numerical Aperture Module
----------------------------------------

VERSION 2.1
This module calculates the fiber NA using a variable threshold method
(presently set at 5%). The input data set may either be the rough
differentiated data, or a smoothed version of the same.
First, determine which data to use to calculate the NA.

Next, locate the 15% points to be sure we are off the noise floor.

Finally, start at the 15% level, and search forward to the next interpolated threshold crossing.

Compute a crossing using linear interpolation.

\[
\text{INDEX} = \text{INDEX} + 1
\]

\[
\text{IF INDEX} > \text{INDEXD}
\]

\[
\text{END WHILE}
\]

\[
\text{INDEXD} = \text{INDEXD} - 1
\]

\[
\text{END IF}
\]

\[
\text{INDEX} = \text{INDEX} + 1
\]

\[
\text{IF INDEX} > \text{INDEXD}
\]

\[
\text{END WHILE}
\]

\[
\text{Sin2} = \text{Sin1} + (\text{Sin1} - \text{Sin2}) \cdot (\text{Threshold} - \text{Sin1}) / (\text{Threshold} - \text{Sin2})
\]

\[
\text{Numaper} = \text{Sin1} - \text{Sin2}
\]

\[
\text{Done:} \text{END}
\]

\[
\text{DIMMS1(0:8)} : R(1518)
\]
The following data statements are for the menu prompts.

**FIRST MENU**
- DATA RUN FIBER TESTS, PRINT PROGRAM LISTING, EXAMINE SYSTEM DATA, EQUIPMENT
- DATA Set Time and Date, Save Results (ARCHIVE), Retrieve Archived Data, Restart Program

**SECOND MENU**
- DATA RETURN TO MAIN MENU, LOAD FIBER AND IDENTIFY, FIBER INPUT ALIGN, FIBER OUTPUT ALIGN
- DATA Fiber Test 1: SPECTRAL ATTENUATION, Fiber Test 2: DIFFERENTIAL MODAL ATTEN, Fiber Test 3/4: FAR FIELD (edge/pinhole)
- DATA Fiber Test 5: NEAR FIELD (Inactive)

**THIRD MENU**
- DATA RETURN TO MAIN MENU, RETURN TO FIBER TEST MENU, LOAD FIBER AND IDENTIFY, Run FAR FIELD-pinhole (low loss fiber)
- DATA Run FAR FIELD-edge (higher loss fiber), Recall data from previous test, not used, not used

**FOURTH MENU**
- DATA "RETURN TO MAIN MENU", "800 nm to 1800 nm (Grating 1)"
- DATA "1800 nm to 2700 nm (Grating 2)", "2700 nm to 4000 nm (Grating 3)"
- DATA "800 nm to 4000 nm (Full Spectral Range)" , "Recall data from previous test"
- DATA "Change wavelength stepping increment", "Enter your own wavelength range"

**FIFTH MENU**
- DATA "RETURN TO MAIN MENU", "800 nm to 1800 nm (Grating 1)"
- DATA "1800 nm to 2700 nm (Grating 2)", "2700 nm to 4000 nm (Grating 3)"
- DATA "800 nm to 4000 nm (Full Spectral Range)" , "Recall data from previous test"
- DATA "Change wavelength stepping increment", "Enter your own wavelength range"
FIRST MENU KEY LABELS, 9817 MAIN MENU

DATA " FIBER TESTS", PROGRAM LISTING, SYSTEM DATA, PRESET EQUIPMENT, SET TIME & DATE, ARCHIVE, RETRIEVE, "RESTART PROGRAM"

SECOND MENU KEY LABELS, 9817 FIBER TESTS MENU

DATA " MAIN MENU", " LOAD FIBER", " INPUT ALIGN", " OUTPUT ALIGN", " SPECTRAL ATTEN", " DIF MODE ATTEN", " FAR FIELD", " NEAR FLD INACTIVE"

THIRD MENU KEY LABELS, 9817 FAR FIELD MENU

DATA " MAIN MENU", " TEST MENU", " LOAD FIBER", " PIN HOLE (silica)", " EDGE (usual)", " RECALL DATA", " "

FOURTH MENU KEY LABELS, 9817 DMA MENU

DATA " MAIN MENU", " GRATING 1", " GRATING 2", " GRATING 3", " GRATINGS 1/2/3", " RECALL DATA", " WAVELEN STEP", " USER DEFINED"

FIFTH MENU KEY LABELS, 9817 SPECTRAL ATTENUATION MENU

DATA " MAIN MENU", " GRATING 1", " GRATING 2", " GRATING 3", " GRATINGS 1/2/3", " RECALL DATA", " WAVELEN STEP", " USER DEFINED"

The following section creates the various menus.

Menu_1: Menu_num = 1

Title$ = "NRL IR FIBER CHARACTERIZATION SYSTEM"

Curr_wave_step = Wave_step

GOSUB Clr_screen

STATUS KBD, Key_id

IF BIT(Key_id, 5) THEN RESTORE Data17

READ K$(*)

GOSUB Clrscreen

The following section creates the various menus.

ON KEY 0 LABEL " GOTO Update_time"

ON KEY 1 LABEL K$(1,1) GOTO Key1_1

ON KEY 2 LABEL K$(1,2) GOTO Key1_2

ON KEY 3 LABEL K$(1,3) GOTO Key1_3

ON KEY 4 LABEL K$(1,4) GOTO Key1_4

ON KEY 5 LABEL K$(1,5) GOTO Key1_5

ON KEY 6 LABEL K$(1,6) GOTO Key1_6

ON KEY 7 LABEL K$(1,7) GOTO Key1_7

ON KEY 8 LABEL K$(1,8) GOTO Key1_8

" GOTO Update_time"
Menu_2: Title$="FIBER TEST MENU"
Menu_num=2
GOSUB Draw_box2
BEEP
ALPHA ON
GRAPHICS ON
ON KEY 0 LABEL "" GOTO Update_time
ON KEY 1 LABEL K$(2,1) GOTO Key2_1
ON KEY 2 LABEL K$(2,2) GOTO Key2_2
ON KEY 3 LABEL K$(2,3) GOTO Key2_3
ON KEY 4 LABEL K$(2,4) GOTO Key2_4
ON KEY 5 LABEL K$(2,5) GOTO Key2_5
ON KEY 6 LABEL K$(2,6) GOTO Key2_6
ON KEY 7 LABEL K$(2,7) GOTO Key2_7
ON KEY 8 LABEL K$(2,8) GOTO Key2_8
ON KEY 9 LABEL "" GOTO Update_time
GOTO Update_time
Menu_3: Title$="FAR FIELD MENU"
Menu_num=3
GOSUB Draw_box3
BEEP
ALPHA ON
GRAPHICS ON
ON KEY 0 LABEL "" GOTO Update_time
ON KEY 1 LABEL K$(3,1) GOTO Key3_1
ON KEY 2 LABEL K$(3,2) GOTO Key3_2
ON KEY 3 LABEL K$(3,3) GOTO Key3_3
ON KEY 4 LABEL K$(3,4) GOTO Key3_4
ON KEY 5 LABEL K$(3,5) GOTO Key3_5
ON KEY 6 LABEL K$(3,6) GOTO Key3_6
ON KEY 7 LABEL K$(3,7) GOTO Key3_7
ON KEY 8 LABEL K$(3,8) GOTO Key3_8
ON KEY 9 LABEL "" GOTO Update_time
GOTO Update_time
Menu_4: Title$="DIFFERENTIAL MODAL ATTENUATION"
Menu_num=4
GOSUB Draw_box4
PRINT
PRINT USING "26X,28A,3D,3A":"Current Wavelength step is:" ;Curr_wave_step ,"nm"
BEEP
ALPHA ON
GRAPHICS ON
ON KEY 0 LABEL "" GOTO Update_time
ON KEY 1 LABEL K$(4,1) GOTO Key4_1
ON KEY 2 LABEL K$(4,2) GOTO Key4_2
ON KEY 3 LABEL K$(4,3) GOTO Key4_3
ON KEY 4 LABEL K$(4,4) GOTO Key4_4
ON KEY 5 LABEL K$(4,5) GOTO Key4_5
ON KEY 6 LABEL K$(4,6) GOTO Key4_6
ON KEY 7 LABEL K$(4,7) GOTO Key4_7
ON KEY 8 LABEL K$(4,8) GOTO Key4_8
ON KEY 9 LABEL "" GOTO Update_time
GOTO Update_time
Menu_5: Title$="SPECTRAL ATTENUATION MENU"
Menu_num=5
GOSUB Draw_box5
A key press from any menu causes the program to branch to a point below:

First menu branches.

Key 1: GOSUB Clr_screen
  GOTO Menu_2

Key 2: GOSUB Clr_screen
  CALL Proglist
  GOTO Menu_1

Key 3: GOSUB Clr_screen
  CALL Systemdata
  CALL Serialno
  GOTO Menu_1

Key 4: GOSUB Clr_screen
  CALL Preset
  GOTO Menu_1

Key 5: GOSUB Clr_screen
  CALL Timeset
  GOTO Menu_1

Key 6: GOSUB Clr_screen
  CALL Inalign
  GOTO Menu_1

Key 7: GOSUB Clr_screen
  CALL Outalign
  GOTO Menu_1

Key 8: GOSUB Clr_screen
  CALL MenuS
  Go to Spectral Attenuation menu

Key 9: GOSUB Clr_screen
  CALL Menu4
  Go to DMA menu

Second menu branches.

Key 1: GOSUB Clr_screen
  GOTO Menu_1

Key 2: GOSUB Clr_screen
  CALL Fiberload
  GOTO Menu_2

Key 3: GOSUB Clr_screen
  CALL Inalign
  GOTO Menu_2

Key 4: GOSUB Clr_screen
  CALL Outalign
  GOTO Menu_2

Key 5: GOSUB Clr_screen
  GOTO Menu_5

Key 6: GOSUB Clr_screen
  GOTO Menu_4

Key 7: GOSUB Clr_screen
  GOTO Menu_3

Key 8: GOSUB Clr_screen
  GOTO Menu_2

Secon second menu branches.

Key 1: GOSUB Clr_screen
  GOTO Menu_1

Key 2: GOSUB Clr_screen
  CALL Fiberload
  GOTO Menu_2

Key 3: GOSUB Clr_screen
  CALL Inalign
  GOTO Menu_2

Key 4: GOSUB Clr_screen
  CALL Outalign
  GOTO Menu_2

Key 5: GOSUB Clr_screen
  GOTO Menu_5

Key 6: GOSUB Clr_screen
  GOTO Menu_4

Key 7: GOSUB Clr_screen
  GOTO Menu_3

Key 8: GOSUB Clr_screen
  GOTO Menu_2
IFourth menu branches.

Key4_1: GOSUB Clear_screen  
Return to main menu

7130  GOTO Menu_1

7132  Key4_2: GOSUB Clear_screen  
Select this wavelength range for next DMA test

7134  Specwaves("800 TO 1798 STEP "&VAL$(Curr_wave_step))

7136  CALL Fibertest2(0)  
Source_flag=0; run new test

7138  GOTO Menu_4

7140  Key4_3: GOSUB Clear_screen

7142  Specwaves("1800 TO 2898 STEP "&VAL$(Curr_wave_step))

7144  CALL Fibertest2(0)  
Source_flag=0; new test

7146  GOTO Menu_4

7148  Key4_4: GOSUB Clear_screen

7150  Specwaves("2700 TO 4000 STEP "&VAL$(Curr_wave_step))

7152  CALL Fibertest2(0)  
Source_flag=0; new test

7154  GOTO Menu_4

7156  Key4_5: GOSUB Clear_screen

7158  Specwaves("800 TO 4000 STEP "&VAL$(Curr_wave_step))

7160  CALL Fibertest2(0)  
Source_flag=0; new test

7162  GOTO Menu_4

7164  Key4_6: GOSUB Clear_screen

7166  Source_flag=FNDatassource  
First determine data source

7168  CALL ClearDisplay  
Clear data query from screen

7170  CALL Fibertest2(Source_flag)  
Review past data

7172  RETURN

7174  Key4_7: GOSUB Clear_screen

7176  Curr_wave_step=FNGetint("Enter new wavelength stepping increment (20-200): ",10,200)

7178  GOSUB Clear_screen

7180  GOTO Menu_4

7182  Key4_8: GOSUB Clear_screen

7184  First=FNGetint("Enter First Wavelength (600-4000 nm): ",600,4000)

7186  Last=FNGetint("Enter Last Wavelength (600-4000 nm): ",600,4000)

7188  IF Last<First THEN

7190  Temp=First

7192  First=Last

7194  Last=Temp

7196  END IF

7198  Specwaves VAL$(First)" TO "$VAL$(Last)" STEP "$VAL$(Curr_wave_step)

7200  GOSUB Clear_screen
7210 KeyS_1: CALL Cleardisplay  
7212 GOTO Menu_1  
7214 —RETURN  
7216 KeyS_2: CALL Cleardisplay  
7218 Specwaves("800 TO 1798 STEP "&VAL$(Curr_wave_step))  
7220 CALL Fibertestl(0)  
7222 GOTO Menu_5  
7224 KeyS_3: CALL Cleardisplay  
7226 Specwaves("1800 TO 2699 STEP "&VAL$(Curr_wave_step))  
7228 CALL Fibertestl(0)  
7230 GOTO Menu_5  
7232 KeyS_4: CALL Cleardisplay  
7234 Specwaves("2700 TO 4000 STEP "&VAL$(Curr_wave_step))  
7236 CALL Fibertestl(0)  
7238 GOTO Menu_5  
7240 KeyS_5: CALL Cleardisplay  
7242 Specwaves("800 TO 4000 STEP "&VAL$(Curr_wave_step))  
7244 CALL Fibertestl(0)  
7246 GOTO Menu_5  
7248 KeyS_6: CALL Cleardisplay  
7250 Source_flag=FNDatasource  
7252 CALL Cleardisplay  
7254 CALL Fibertestl(Source_flag)  
7256 RETURN  
7258 KeyS_7: CALL Cleardisplay  
7260 Curr_wave_step=FNGetint("Enter new wavelength stepping increment (20-2000): ",10,200)  
7262 CALL Cleardisplay  
7264 GOTO Menu_5  
7266 KeyS_8: CALL Cleardisplay  
7268 First=FNGetint("Enter First Wavelength (600-4000 nm): ",600,4000)  
7270 Last=FNGetint("Enter Last Wavelength (600-4000 nm): ",600,4000)  
7272 IF Last<First THEN  
7274 Temp=First  
7276 First=Last  
7278 Last=Temp  
7280 END IF  
7282 Specwaves(&VAL$(First)&" TO ",&VAL$(Last)&" STEP ",&VAL$(Curr_wave_step))  
7284 CALL Cleardisplay  
7286 CALL Fibertestl  
7288 GOTO Menu_5  
7290 !This part of the subroutine clears the screen:  
7292 !  
7294 Clr_screen:  
7296 OFF KEY  
7300 DISP " "  
7302 OUTPUT KBD USING ",K","K"  
7304 GCLEAR  
7306 RETURN  
7308 !This part of subroutine prints the current time and date on the menu:  
7310 !  
7312 Update_time:Date$=FNTimedate$  
7314 CONTROL CRT,115  
7316 CONTROL CRT,0:65  
7318 CONTROL CRT,0:65  
7320 OUTPUT CRT;Date$[1,POS(Date$," ")]  
7322 CONTROL CRT,116  
7324 CONTROL CRT,0:65  
7326 OUTPUT CRT;Date$[POS(Date$," ")+1,LEN(Date$)]  
7328 GOTO Update_time
7340 WINDOW 0,100*RATIO,0,100
7342 FOR Delta=0 TO .8 STEP .8
7344 MOVE Delta*RATIO/1.3,12+Delta
7346 DRAW Delta*RATIO/1.3,92-Delta
7348 DRAW 40*RATIO/1.3,92-Delta
7350 MOVE 0,-3
7352 IDRAW 0,8
7354 IDRAW 111*RATIO/1.3,0
7356 IDRAW 0,-8
7358 IDRAW -111*RATIO/1.3,0
7360 MOVE 121*RATIO/1.3,92-Delta
7362 DRAW (130-Delta)*RATIO/1.3,92-Delta
7364 DRAW (130-Delta)*RATIO/1.3,12+Delta
7366 DRAW Delta*RATIO/1.3,12+Delta
7368 NEXT Delta
7370 CSIZE 5,60
7372 FOR Delta=0 TO .3 STEP .2
7374 MOVE 10.5*RATIO/1.3,90
7376 MOVE Delta*RATIO/1.3,0
7378 LABEL Title$
7380 NEXT Delta
7382 ! Time, date, and title:
7384 CONTROL CRT,1;5
7386 CONTROL CRT,0;59
7388 OUTPUT CRT;"DATE:"
7390 CONTROL CRT,1;6
7392 CONTROL CRT,0;59
7394 OUTPUT CRT;"TIME:"
7396 CONTROL CRT,1;7
7398 OUTPUT CRT USING Headimage
7400 CONTROL CRT,1;9
7402 GOTO Box_end
7404 !
7406 Draw_box2:
    !Draw background for Fiber Tests Menu
7408 GINIT
7410 WINDOW 0,100*RATIO,0,100
7412 FOR Delta=0 TO .8 STEP .8
7414 MOVE Delta*RATIO/1.3,12+Delta
7416 DRAW Delta*RATIO/1.3,92-Delta
7418 DRAW 40*RATIO/1.3,92-Delta
7420 MOVE 0,-3
7422 IDRAW 0,8
7424 IDRAW 50*RATIO/1.3,0
7426 IDRAW 0,-8
7428 IDRAW -50*RATIO/1.3,0
7430 MOVE 90*RATIO/1.3,92-Delta
7432 DRAW (130-Delta)*RATIO/1.3,92-Delta
7434 DRAW (130-Delta)*RATIO/1.3,12+Delta
7436 DRAW Delta*RATIO/1.3,12+Delta
7438 NEXT Delta
7440 FOR Delta=0 TO .3 STEP .05
7442 CSIZE 7
7444 MOVE 0.93
7446 MOVE Delta*RATIO/1.3,Delta
7448 LABEL "FOA-2000"
7450 CSIZE 4
7452 MOVE 95.93
7454 MOVE Delta*RATIO/1.3,Delta/10
7456 LABEL "PK - VPI"
7458 NEXT Delta
7460 CSIZE 5.58
7462 FOR Delta=0 TO .3 STEP .2
I Time, date, and title:
7472 CONTROL CRT,115
7474 CONTROL CRT,0159
7476 OUTPUT CRT; "DATE:"
7478 CONTROL CRT,116
7480 CONTROL CRT,0159
7482 OUTPUT CRT; "TIME:"
7484 CONTROL CRT,117
7486 OUTPUT CRT USING Headimage
7488 CONTROL CRT,119
7490 CONTROL CRT,119
7492 GOTO Box_end
7494 !
7496 Draw_box3:
7498 GINIT
7500 WINDOW 0,100*RATIO,0,100
7502 FOR Delta=0 TO .8 STEP .8
7504 MOVE Delta*RATIO/1.3,12+Delta
7506 DRAW Delta*RATIO/1.3,92-Delta
7508 DRAW 42*RATIO/1.3,92-Delta
7510 IMOVE 0,-3
7512 IDRAW 0,8
7514 IDRAW 48*RATIO/1.3,0
7516 IDRAW 0,-8
7518 IDRAW -48*RATIO/1.3,0
7520 MOVE 90*RATIO/1.3,92-Delta
7522 DRAW (130-Delta)*RATIO/1.3,92-Delta
7524 DRAW (130-Delta)*RATIO/1.3,12+Delta
7526 DRAW Delta*RATIO/1.3,12+Delta
7528 NEXT Delta
7530 CSIZE 5,.58
7532 FOR Delta=0 TO .3 STEP .2
7534 MOVE 45.0*RATIO/1.3,90
7536 IMOVE Delta*RATIO/1.3,90
7538 LABEL Title$ NEXT Delta
7540 !
7542 Time, date, and title:
7544 CONTROL CRT,115
7546 CONTROL CRT,0159
7548 OUTPUT CRT; "DATE:"
7550 CONTROL CRT,116
7552 CONTROL CRT,0159
7554 OUTPUT CRT; "TIME:"
7556 CONTROL CRT,116
7558 OUTPUT CRT USING Headimage
7560 CONTROL CRT,118
7562 GOTO Box_end
7564 !
7566 Draw_box4:
7568 GINIT
7570 WINDOW 0,100*RATIO,0,100
7572 FOR Delta=0 TO .8 STEP .8
7574 MOVE Delta*RATIO/1.3,12+Delta
7576 DRAW Delta*RATIO/1.3,92-Delta
7578 DRAW 20*RATIO/1.3,92-Delta
7580 IMOVE 0,-3
7582 IDRAW 0,8
7584 IDRAW 95*RATIO/1.3,0
7586 IDRAW 0,-8
7588 IDRAW -95*RATIO/1.3,0
7590 MOVE 115*RATIO/1.3,92-Delta
7592 DRAW (130-Delta)*RATIO/1.3,92-Delta
7594 DRAW (130-Delta)*RATIO/1.3,12+Delta
7596 DRAW Delta*RATIO/1.3,12+Delta
7598 !
MOVE 24.0*RATIO/1.3,90
IMOVE Delta*RATIO/1.3,0
LABEL Titles

NEXT Delta

Time, date, and title:
CONTROL CRT,1;S
OUTPUT CRT;"DATE:"
CONTROL CRT,1;S
OUTPUT CRT;"TIME:"
CONTROL CRT,1;S
OUTPUT CRT USING Headimage
CONTROL CRT,1;S
GOTO Box_end

Draw_box5:

GINIT
WINDOW 0,100*RATIO,0,100
FOR Delta=0 TO .8 STEP .8
MOVE Delta*RATIO/1.3,12+Delta
DRAW Delta*RATIO/1.3,92-Delta
DRAW 25*RATIO/1.3,92-Delta
IMOVE 0,-3
IDRAW 0,8
IDRAW 77*RATIO/1.3,0
IDRAW 0,-8
IDRAW -77*RATIO/1.3,0
MOVE 102*RATIO/1.3,92-Delta
DRAW (130-Delta)*RATIO/1.3,92-Delta
DRAW (130-Delta)*RATIO/1.3,12+Delta
DRAW Delta*RATIO/1.3,12+Delta
NEXT Delta

CSIZE 5,.58
FOR Delta=0 TO .3 STEP .2
MOVE 26.5*RATIO/1.3,90
IMOVE Delta*RATIO/1.3,0
LABEL Titles

NEXT Delta

Time, date, and title:
CONTROL CRT,1;S
CONTROL CRT,0;S
OUTPUT CRT;"DATE:"
CONTROL CRT,1;S
OUTPUT CRT;"TIME:"
CONTROL CRT,1;S
OUTPUT CRT USING Headimage
CONTROL CRT,1;S
GOTO Box_end

FOR I=1 TO 4
IF BIT(Key_id,5) THEN
OUTPUT CRT USING Keyimage;I,M$(Menu_num,I)
ELSE
OUTPUT CRT USING Keyimage;I,M$(Menu_num,I)
END IF
NEXT I

FOR I=5 TO 8
OUTPUT CRT USING Keyimage;I,M$(Menu_num,I)
NEXT I
SUB Serialno
  See Machine Serial Numbers
  COM /Sysdata/ Serial_num#,Lasers(*),Filter_flag,Filter(*),Num_focus,Focu
s(*),Cutoff,Low_wave,High_wave,Det_switch
  OUTPUT KBD USING "*,K":"
  PRINT TABXY(5,10):" Machine Serial Number: ":Serial_num$
  ON KEY S LABEL "PROCEED" GOTO Done
  Waiter: GOTO Waiter
  Done: SUBEND

FNGetint(Prompt$, Lo, Hi)
FNGetint: for inputting integer values
  INTEGER Value,I
  DIM Inp$[80]
  Prompt$ = Prompt$ + ",Lo:"
  IF LEN(Inp$)>S OR LEN(Inp$)=0 THEN GOTO Bad inp
  IF LEN(Inp$)=S AND Inp$="32767" THEN GOTO Bad inp
  I=1
  WHILE (I<=LEN(Inp$))
    IF Inp$[I;1]<"0" OR Inp$[I;1]="9" THEN GOTO Bad inp
    I=I+1
  END WHILE
  Value=VAL(Inp$)
  IF Value<Lo OR Value>Hi THEN GOTO Bad inp
  RETURN Value

Bad inp: PRINT "You must enter an integer value between ":Lo:
  PRINT "and ":Hi", inclusive."
  PRINT
  GOTO Get_it
  FNEND

DEF FNgrating(Wavelen)
FNgrating(Wavelen): This determines which grating is required for a wavelength specified in the calling routine.
  INTEGER I
  FOR I=Gratings(0) TO 1 STEP -1
    IF Gratings(I)<=Wavelen THEN RETURN (I)
  NEXT I
  RETURN (-1)! Unknown grating setting
  FNEND
REQUEST ALIGNMENT ROUTINE

This routine gives the user the option of bypassing the alignment.

DISP "Do the fiber ends need to be aligned?"
BEEP
ON KEY 1 LABEL "YES" GOTO Align
ON KEY 5 LABEL "NO" GOTO Done

This routine gives the user the option of bypassing the alignment.

Do the fiber ends need to be aligned?

BEEP
ON KEY I LABEL "YES" GOTO Align
ON KEY S LABEL "NO" GOTO Done

Infinite: GOTO Infinite
Align: OFF KEY
OUTPUT KBO USING ",K",K"
DISP
CALL Fibertype
CALL Inalign
CALL Outalign

Done: OFF KEY
DISP
OUTPUT 1<80 USING "t.K"; "K"

SUBEND

SUB Init_foa_cntrl

FOA-2600 new commends for IR detectors and multi-grating monochromator

COM /lopaths/ @Foa2000, @Egg5205, @Tek7514, @Bncdelay, Printer_add
COM /Addition/ Cur_r_wave, Gratings( ), Cur_grating, Wave_step

CALL F200send("INZ SEL -4000 -RANGE -4000 +RANGE >MTAB",1)

NOTE: the routines below are written in FORTH, the native operating
system for the Z-80 processor card inside the FOA-2000 control box.
For more information, see a book on the FORTH language.

Change the monochromator's motor table to reflect the different mono.
'M+' changes the motor step routine's address to run the mono in
the other direction.

MPY 1 is the multiplicative display scaling factor.
DIV 1 is the display scaling divisor. DIV will change with the
selected gratings (see the GRAT? commands below). This one is
the value for grating #1.

-32000 FTARG is the distance to look backwards (in motor steps)
for the optical sensor edge when trying to locate the monochrom's
zero order window.

MTAB copies the data into the permanent motor table
-32700 -RANGE allows the moves in complete range forward and back.

CALL F200send("WAV SEL 'M+ DIR ' 'MPY 'DIV -1200 FTARG MTAB ",1)
CALL F200send("WAV SEL -32700 -RANGE -32700 +RANGE 'SENS 'MTAB",1)

0 SCX allows us to define the new commands below
CALL F2000send("0 SCX ",1)

Since the computer does not yet know what grating the monochromator
is turned to, set current wavelength to -1 (unknown)

Cur_grating=0
Cur_wave=-1

Define the gratings installed in the system.
Gratings(2)=1800 | grating 2 for >1800 nm but <2700 nm
Gratings(3)=2700 | grating 3 for >2700 nm

Set default wavelength step for spectral attenuation & diff. modal attenuation.
Wave_step=10

The new command MARKSTART is a dummy to mark where the new commands
start in RAM. If this routine has already been called, then we
will recover the RAM already used by FORGETting the defined commands
and re-defining them. If this routine hasn't been called yet, then
FORGET MARKSTART will produce an error. This is OK, but we can't
use the F2000send routine, since it will trap the error.

OUTPUT @Foa2000:"FORGET MARKSTART"

IF BIT(Statbyt,4) THEN Wait
CALL F2000send(": MARKSTART ",1)

This is the zero order find routine for the monochromator
CALL F2000send(" : FINDBG LOC @ 60 OVER +/- GOTO @ LOC ! FTARG @ DARK IF MER7 THEN LOC @ ",1)
CALL F2000send(" : FINDBG LOC @ 40 LIGHT IF -40 LIGHT IF MER7 THEN THEN FOLY @ MTDY @ FINDBG ",1)
CALL F2000send(" : OSEEK WAV SEL FINDBG @ FOUND ! ",1)

These are the commands to set the controller to understand the grating
it's trying to run. GRAT 1 is the command for grating #1, etc.

CALL F2000send(" : GRAT1 WAV SEL 12 DIV @ MTAB ",1)
CALL F2000send(" : GRAT2 WAV SEL 5 DIV @ MTAB ",1)
CALL F2000send(" : GRAT3 WAV SEL 3 DIV @ MTAB ",1)

The command CVTLOC converts motor steps into wavelength. THIS SHOULD
NOT BE USED OVER GPIB!!!

CALL F2000send(" : CVTLOC DIV @ OUP 0= IF DROP ELSE / THEN MPY @ DUP 0= IF DROP ELSE * THEN ",1)

The command CUTWAVE converts wavelength to motor steps. THIS SHOULD
NOT BE USED OVER GPIB!!!

CALL F2000send(" : CUTWAVE MPY @ DUP 0= IF DROP ELSE / THEN DIV @ DUP 0= IF DROP ELSE * THEN ",1)

The command GETNEARWAV moves the mono near the wavelength desired.
If the wavelength is too far away, it will need to be called
more than once. If an error occurs, GETNEARWAV returns either 1
or -1. If it needs to be called again, it returns 0. If it doesn't
need to be called again, it will return -88.
THIS COMMAND SHOULDN'T BE USED OVER GPIB!!!

CALL F2000send(" : GETNEARWAV LOC @ CUTLOC - DUP ABS 650 > IF LOC @ CUTLOC C SWP 0< IF 650 - ELSE ",1)
CALL F2000send(" : GETNEARWAV LOC @ CUTLOC DUP DUP 32000 > IF DROP DROP 1 ELSE - 2000 < IF DROP 1 ELSE ",1)
CALL F2000send(" : MOV IF MER7 ELSE 0 THEN THEN T " : F DROP -89 THEN "}
These two GPIB commands, \texttt{GERMAIN} and \texttt{INSB}, select one of the two detectors on the bench:

\begin{verbatim}
CALL F2000send(":: GERMAIN HIGH ",1)
CALL F2000send(":: INSB SILICON APDET ",1)
\end{verbatim}

\begin{verbatim}
CALL F2000send(":: DELAY 0 DO 255 0 DO LOOP LOOP ",1)
CALL F2000send(":: GREL BS1 SEL 1000 MINDLY ! LOC @ 3 AND ",1)
CALL F2000send(":: GREL BS1 SEL MINDLY ! LOC @ 3 AND ",1)
CALL F2000send(":: TURN 70 1000 GREL 432 50 GREL 32 DELAY -40 1000 GREL ",1)
\end{verbatim}

\begin{verbatim}
CALL F2000send(":: T01 WAV SEL 24 DIV ! MTAB ",1)
\end{verbatim}

\begin{verbatim}
SUB Ffnormalize(Dataflag$)

<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Ffsmoothdata(I,1)=Ffsmoothdata(I,1)/Maxval.

NEXT I

END IF

SUBEND

SUB Ffdiff

! Ffdiff: Differentiates integrated farfield pattern that is 
! derived by the knife-edge technique (see note below).

This routine is used to differentiate the farfield pattern with respect 
! to sin(theta). Note that differentiation should actually be with 
! respect to the vertical scanner position, but the above method is 
! equivalent (and simpler) because there is a linear relationship between 
! the scanner position and sin(theta), and we are not interested in the 
! magnitude after differentiation since we will normalize anyway.

Also note that this routine takes the negative derivative due to the 
! physical motion of the farfield scanner (see the RAW data plot).

COM /Fftempdata/ Ffrawdata(*),Ffdiffdata(*),Ffsmoothdata(1)

INTEGER I

REAL Sintheta

CALL Rundisplay("Differentiating far-field pattern.")

FOR I=2 TO Ffrawdata(0,0)

Ffdiffdata(I-1,0)=Ffrawdata(I,0)

Ffdiffdata(I-1,1)=-(Ffrawdata(I,1)-Ffrawdata(I-1,1))/(Ffrawdata(I,0)-

Ffrawdata(I-1,0))

NEXT I

Ffdiffdata(0,0)=Ffrawdata(0,0) ! Reduce the number of points by 1

SUBEND

SUB Ffcorrect

! FFCorrect: This corrects far-field measurements for COS(PHI)

COM /Fftempdata/ Ffrawdata(*),Ffdiffdata(*),Ffsmoothdata(*)

INTEGER I

REAL Sintheta

CALL Rundisplay("Correcting far-field pattern.")

FOR I=1 TO Ffdiffdata(0,0)

Sintheta=Ffdiffdata(I,0)

Ffdiffdata(I,1)=Ffdiffdata(I,1)*SQRT(1-Sintheta*Sintheta)

NEXT I

SUBEND

DEF FNGetffwave

FNGetwave: this function asks the operator for the wavelength for,
! the farfield scan.

COM /Sysdata/ Serial_num$(40),Laser(*),Filter_flag,Filter(*),Num_focus
!,Focus(*),Cutoff,Low_wave,High_wave,Det_switch

FOR I=1 TO Ffrawdata(0,0)

Ffdiffdata(I,0)=Ffrawdata(I,0)

Ffdiffdata(I,1)=Ffrawdata(I,1)*SQRT(1-Sintheta*Sintheta)

NEXT I

SUBEND

SUB FNGetwave

FOR I=1 TO Ffrawdata(0,0)

Ffdiffdata(I,0)=Ffrawdata(I,0)

Ffdiffdata(I,1)=Ffrawdata(I,1)*SQRT(1-Sintheta*Sintheta)

NEXT I

SUBEND
SUB Align(Axis$, Step_len, Fail_flag, Min_factor, Accuracy)

Fiber Alignment Module

3/8/90

This module is called by both Inalign and Outalign. It is responsible for the alignment of a single axis only (specified in the call).

COM /Align_read/ Reading

REAL Signal

INTEGER Position, Step

Step=ABS(Step_len)

Maxallowed=.3

Atte0 'Starting attenuator setting

PRINTER IS CRT

PRINT " "

PRINT " "

PRINT USING "7X,7A,6A,3X,14A,4D";"AXIS =",Axis$","Step length =",Step_len

PRINT " "

PRINT USING "4X,10A,9X,6A,8X,8A";"status","signal","position"

Define axis and initialize:

CALL F2000send(Atis$[POS(Axis$,"-")-1]&A .is$[POS(Axis$,"-")]+11]&" COU PL"

Start: Position=0 'Come back here only if signal greater than Maxallowed.

Reading=FNVoltmeter(Accuracy) 'Get a reading from the S205/7

PRINT USING "4X,10A,8X,M,DDDDDD,8X,DDDD.D";"INITIAL",Reading,Position

IF Reading>Maxallowed THEN 'Make sure S205/7 isn't saturated

GOSUB Set_attn

GOTO Start

END IF

Signal1=Reading 'Store initial signal before moving

Position1=Position 'and initial position in case it's max

Position=Position+Step_len 'Increment position by step length

IF Position<-800 OR Position>800 THEN GOTO Failure 'Keep in range

CALL F2000send(VAL$(Position)&" &Axis$,1)

ELSE

CALL F2000send(VAL$(Position)&" &Axis$,1)

ENDIF

IF Reading Signal1 THEN

PRINT " Reversing direction"

Step_len=-Step_len 'Step in the other direction

Position=Position+Step_len 'Move back to original position

ENDIF

ELSE

PRINT " Direction okay"

Signal1=Reading 'Store a new max value

Position1=Position 'and corresponding position

ENDIF
CALL F2000send(VAL$(Position)&" &Axis$,1)

! Now start looking for the signal to begin decreasing again, indicating that we have passed the maximum level.

Reading=FNVoltmeter(Accuracy)  ! Get a reading from the 5205
IF Reading>Maxallowed THEN  ! Check again for saturation
GOSUB Set_attn
GOTO Start
END IF
PRINT USING "4X,10A,8X,M.DDDDDD,8X,DDDD.D";"MAX SEARCH",Reading,Position
IF Reading>Signall THEN  ! Signal still increasing
PRINT " Signal still increasing"
Signall=Reading  ! And put the new level in register
Position1=Position  ! As well as its position
GOTO Loop!
END IF

! If signal is decreasing, keep moving past the peak until signal is some percentage of the max value to avoid peaking on a noise spike:
IF Reading>Min_factor*Signall THEN GOTO Loop!
CALL F2000send(VAL$(Position1)&" &Axis$,1)  ! Move to max position
Reading=FNVoltmeter(Accuracy)  ! Re-confirm max signal
PRINT USING "4X,10A,8X,M.DDDDDD,8X,DDDD.D";"FINAL",Reading,Position!
CALL Setscale(Accuracy,Reading)
BEEP
CALL Cleardisplay
SUBEXIT
Set_attn:Att=Att+1  ! Change attenuator to reduce signal
IF Att>4 THEN  ! Have we run out of range?
BEEP
OUTPUT KBD USING ";K","K"
GCLEAR
CONTROL CRT,1;10
OUTPUT CRT;"ALIGNMENT DIFFICULTIES"
OUTPUT CRT;"Signal greater than "&VAL$(Maxallowed)" volts; too great for proper alignment."
Hang_over: GOTO Hang_over
ELSE
CALL F2000send(VAL$(INT(Att))&" ATTENUAT",1)
END IF
GOTO Start
ELSE
Failure: Fail_flag=1
SUBEXIT
SUBEND
SUB Cleardisplay
OUTPUT KBD USING "K","K"
GCLEAR
SUBEND
SUB Steptest(Axis$)
SUBEND
8662 | All parameters should be specified in motor steps, which are twice as large as displayed on the front panel for the x and y axes, and are in the ratio of 10:8 larger for the z axis. Front panel reads in microns.

8671 | CALL F2000send(VAL$(0)&" ATTENUAT",l)
8673 | CALL F2000send("XMIT CHOP-ON SPOT-IN")
8675 | CALL F2000send("GERMAIN VOUT FF-OUT TARGET-OUT")
8676 | Step=50
8678 | PRINT " 
8680 | PRINT " AXIS =",Axis$
8682 | CALL F2000send(Axis$[1,POS(Axis$,"-")-1]&Axis$[POS(Axis$,"-")+1,1]" COUL")
8684 | PRINT " 
8686 | PRINT USING "10A,2X,10A";"POSITION","SIGNAL"
8688 | Position=700
8690 | CALL F2000send(VAL$(Position)&" &Axis$,l)
8692 | Hang_it: GOTO Hang_it
8694 | Loopsy: IF Position<=600 THEN
8696 | Reading=FNVoltmeter(.1)
8698 | PRINT USING "DOOD,D,5X,M.DDDDDDD";Position,Reading
8700 | Position=Position+Step
8702 | CALL F2000send(VAL$(Position)&" &Axis$,l)
8704 | GOTO Loopsy
8706 | END IF
8708 | Position=0
8710 | CALL F2000send(VAL$(Position)&" &Axis$,l)
8712 | PRINT IS CRT
8714 | SUBEND
8716 | SUB Inalign
8718 | INPUT AUTO-ALIGNMENT MODULE
8720 | COM /Align_param/ Ap(*) | Auto-alignment parameters set by FIBERTYPE
8722 | R700 COM /Align_read/ Reading
8724 | REAL Sig_change
8726 | INTEGER Trial_no
8728 | DIM Sig(10)
8730 | CALL Rundisplay("Input Auto-Alignment in progress.")
8732 | CALL to see if the fiber type has been set.
8734 | IF Ap(0)=0 THEN CALL Fibertype
8736 | Initialization parameters:
8738 | Rough_dx=Ap(1) | Step size for rough alignment
8740 | Rough_dy=Ap(2)
8742 | Rough_dz=Ap(3)
8744 | Fine_dx=Ap(4) | Step size for fine alignment
8746 | Fine_dy=Ap(5)
8748 | Fine_dz=Ap(6)
8750 | Rough_min=.95 | Search past the peak for this percent of max. power
8752 | Fine_min=.98 | Same for fine (change in conjunction with accuracy)
8754 | Rough_acc=.2 | Accuracy used in calling EG&G in rough align
8756 | Fine_acc=.1 | Accuracy for fine (change with Fine_min)
IF Rough_dx=20 THEN PRINT TABXY(5,6);"Fiber diameter of 50 microns is assumed."
IF Rough_dx=36 THEN PRINT TABXY(5,6);"Fiber diameter of 85 microns is assumed."
IF Rough_dx=40 THEN PRINT TABXY(5,6);"Fiber diameter of 100 microns is assumed."
IF Rough_dx=60 THEN PRINT TABXY(5,6);"Fiber diameter of 150 microns is assumed."

WAIT 2
CALL Rundisplay(" ")
!
Begin the alignment loop. Come back in the event of failure in z.

Retry: !First initialize loop parameters and set up system
OFF KEY
Trial_no=Trial_no+1
Fail_flag=0
PRINT TABXY(60,10);
PRINT USING "10A,OO";"Inalign #",Trial_no
CALL F2000send("XMIT LED LED-ON CHOP-ON SPOT-IN")
CALL F2000send("GERMAIN V20 FF-OUT TARGET-OUT")
CALL F2000send("STAGE0",1)
!
Rough align each axis. After each alignment call, check the alignment parameter. If it fails, do the alignment manually.
!
CALL Align("IN-X",Rough_dx,Fail_flag,Rough_min,Rough_acc)
IF Fail_flag=1 THEN Failure
CALL Align("IN-Y",Rough_dy,Fail_flag,Rough_min,Rough_acc)
IF Fail_flag=1 THEN Failure
CALL Align("IN-Z",Rough_dz,Fail_flag,Rough_min,Rough_acc)
IF Fail_flag=1 THEN Failure_z
CALL F2000send("STAGE0",1)
!
Now, fine align each axis. Again, test alignment parameters and do the alignment manually if any parameters are not met.
!
CALL Align("IN-X",Fine_dx,Fail_flag,Fine_min,Fine_acc)
IF Fail_flag=1 THEN Failure
CALL Align("IN-Y",Fine_dy,Fail_flag,Fine_min,Fine_acc)
IF Fail_flag=1 THEN Failure
CALL Align("IN-Z",Fine_dz,Fail_flag,Fine_min,Fine_acc)
IF Fail_flag=1 THEN Failure_z
CALL F2000send("STAGE0",1)
!
Test to see how repeatable the alignment is. If there is more than 1% difference in signal between alignments, give user the choice to retry.
!
Sig(Trial_no)=Reading
IF Trial_no<2 THEN !Do alignment at least twice, test for stability
GOTO Retry
ELSE
Sig_change=100*(Sig(Trial_no)-Sig(Trial_no-1))/Sig(Trial_no-1)
IF Sig_change>1 THEN
PRINT USING "22A,MOD.D,22A,DD,4A,DD";"A change in signal of ",Sig_change," occurred between INALIGN trial",Trial_no," and",Trial_no-1
PRINT "Press f1 to RERUN the alignment routine, f5 to EXIT."
ON KEY 1 LABEL " RERUN" GOTO Retry
ON KEY 5 LABEL " EXIT" GOTO Cleanout
Snooze_dude: GOTO Snooze_dude
END IF
END IF

If any of the success parameters are not met, this manual alignment routine is entered to give the user manual control of the FOA-2000 and prompt him to manually align the fiber. Failure_z anticipates particular errors which result from non-optimal placement of the fiber in the vacuum chuck. The user is prompted to focus the fiber end at a position particular to this system figured to encounter the least error.

Failure_z:BEEP

IF Trial_no>1 THEN GOTO Failure

CALL F2000send("ALIGN INZ COUPL 3000 DARK","I")  "Find edge of INZ sensor
CALL F2000send("INZ ZER -300 GOTO INZ ZER","I")  "Back up and stop
CALL F2000send("250 IN-Z","I")

PRINT TABXY(1,17),"INALIGN -- Unsuccessful auto-alignment."
PRINT TABXY(1,18),"Adjust the input end of the fiber in the vacuum chuck until"
PRINT TABXY(1,19),"it comes into rough focus on the monitor. Then press RE-TRY."

ON KEY 5 LABEL "PROCEED" GOTO Quit
ON KEY 6 LABEL "RE-TRY AUTO" GOTO Retry_prep

Wait_here: GOTO Wait_here

Failure: OFF KEY
CALL Cleardisplay
PRINT TABXY(1,17),"INALIGN -- Auto-alignment unsuccessful in the IN-Z monitor."
PRINT TABXY(1,18),"Align input end of fiber using the FOA-2000 panel controls."

ON KEY 5 LABEL "PROCEED" GOTO Quit
ON KEY 6 LABEL "RE-TRY AUTO" GOTO Retry_prep

Wait_here: GOTO Wait_here

Retry_prep: I
OFF KEY
CALL Cleardisplay
CALL Rundisplay("Input Auto-Alignment in progress.")
GOTO Retry

Quit: I
OFF KEY
OUTPUT KBD USING ",K","K"
GCLEAR

SUB Outalign
REAL Sig_change
INTEGER Trial_no
DIM Sig(10)

COM /Align_param/ Ap(*)  "Auto-alignment parameters set by FIBERTYPE
COM /Align_read/ Reading
IF Ap(0)=0 THEN CALL Fibertype

Initialize parameters:

Rough_dx=Ap(1)  \text{Step size for rough alignment}
Rough_dy=Ap(2)
Rough_dz=Ap(3)
Fine_dx=Ap(4) \text{Step size for fine alignment}
Fine_dy=Ap(5)
Fine_dz=Ap(6)

Rough_min=.95 \text{Search past the peak for this percent of max power}
Fine_min=.98 \text{Same for fine (change in conjunction w/accuracy)}
Rough_acc=.2 \text{Accuracy used in calling EG&G in rough align}
Fine_acc=.1 \text{Accuracy for fine (change w/Fine_min)}

Remind the user what your fiber type is.

IF Rough_dx=30 THEN PRINT TABXY(5,6);"Fiber diameter of 50 microns is assumed."
IF Rough_dx=36 THEN PRINT TABXY(5,6);"Fiber diameter of 85 microns is assumed."
IF Rough_dx=40 THEN PRINT TABXY(5,6);"Fiber diameter of 100 microns is assumed."
IF Rough_dx=60 THEN PRINT TABXY(5,6);"Fiber diameter of 150 microns is assumed."

WAIT 2
CALL Rundisplay(""
CALL F2000send("XMIT LED LED-ON CHOP-ON SPOT-OUT")
CALL F2000send("GERMAIN VOUT FF-OUT TARGET-IN")
CALL F2000send("STAGE0",1)

Rough align each axis. After each alignment call, check the alignment parameter. If it fails, do the alignment manually.

Now, fine align each axis. Again, test alignment parameters and do the alignment manually if any parameters are not met.
IF Trial_no<2 THEN
!Do alignment at least twice, test for stability
ELSE
\[ \text{Sig\_change} = 100 \times \frac{\text{Sig}(\text{Trial\_no}) - \text{Sig}(\text{Trial\_no-1})}{\text{Sig}(\text{Trial\_no-1})} \]
PRINT USING "2A,MOD.D,22A,DD,4A,DD"; "A change in signal of \text{"Sig\_change"}, occurred between OUTALIGN trial\_no, \text{"Trial\_no"}, and \text{"Trial\_no-1"}
PRINT "Press f1 to RERUN the alignment routine, f5 to EXIT."
ON KEY 1 LABEL " RERUN" GOTO Retry
ON KEY 5 LABEL " EXIT" GOTO Cleanout
Snooze\_man: GOTO Snooze\_man
END IF
END IF

Now clean up and quit.

PRINT TABXY(16,12),"Output fiber end successfully aligned."
Cleanout: OFF KEY  BEEP
BEEP
WAIT 2
CALL Cleardisplay
SUBEXIT

If any of the success parameters are not met, this manual alignment routine is entered to give the user manual control of the FOA-2000 and prompt him to manually align the fiber. Failure\_z anticipates particular errors which result from non-optimal placement of the fiber in the vacuum chuck. The user is prompted to focus the fiber end at a position particular to this system figured to encounter the least error.

Failure\_z: BEEP
IF Trial\_no>1 THEN GOTO Failure
CALL F2000send("ALIGN OUTZ COUPL 3000 DARK",1) !Find edge of OUTZ sensor
CALL F2000send("OUTZ ZER -900 GOTO OUTZ ZER",1) !Back up and stop
CALL F2000send("250 OUT-2",1)
CALL Cleardisplay
PRINT TABXY(1,17),"OUTALIGN -- Auto-alignment unsuccessful in the OUT-Z motor."
PRINT TABXY(1,18),"Adjust the output end of the fiber in the vacuum chuck until" PRINT TABXY(1,19),"it comes into rough focus on the monitor. Then press RE-TRY."
ON KEY 6 LABEL "PROCEED" GOTO Quit
ON KEY 5 LABEL " PROCEED" GOTO Quit
ON KEY 6 LABEL " RE-TRY AUTO" GOTO Retry\_prep
Wait\_here: GOTO Wait\_here

Failure: OFF KEY
CALL Cleardisplay
PRINT TABXY(1,17),"OUTALIGN -- Unsuccessful auto-alignment."
PRINT TABXY(1,18),"Align output end of fiber using the FOA-2000 panel controls."
ON KEY 5 LABEL "PROCEED" GOTO Quit
ON KEY 6 LABEL " RE-TRY AUTO" GOTO Retry\_prep
Wait\_there: GOTO Wait\_there

Retry\_prep:!
OFF KEY
CALL Cleardisplay
CALL Rundisplay("Output Auto-Alignment in progress.")
GOTO Retry

Now clean up and quit.
PRINT TABXY(1,17),"Output fiber end successfully aligned."

I
SUB NextWave(Wavelen)

| GET NEXT WAVELENGTH MODULE |

VERSION 2.1 IR

COM /Sysdata/ Serial_num$,Lasers(*),Filter_flag,Filters(*),Num_focus,Foc us(*)\, Cutoff,Low_wave,High_wave,Det_switch

COM /Addition/ Curr_wave,Gratings(*),Cur_grating,Wave_step

INTEGER Index

DIM Cmd$(80),Dum$(40)

Cmd$=""

First, make sure the wavelength called is not out of range.

IF Wavelen<Low_wave OR Wavelen>High_wave THEN GOTO Filter_err

Next, figure out which grating to use.

New_grating=FGrating(Wavelen)

If we don't know what wavelength we were at, which grating we were using, or the grating we want isn't the grating currently in use, we will ask the user to switch the grating by hand.

IF Curr_wave=-1 THEN

DISP "PLEASE TURN TO GRATING NUMBER";New_grating

Cur_grating=New_grating

BEEP

ON KEY S LABEL "PROCEED" GOTO Proceed

Infinite: GOTO Infinite

Proceed:

OFF KEY

CALL F2000send("ITI1 300 WAVE",1) !Eliminate possible backlash

OUTPUT KBD USING ";K";

DISP !Clear grating request off screen

END IF

IF Curr_wave=-1 OR Cur_grating<>New_grating THEN

OUTPUT Dum$ USING ";";GRAT";O,""","",#";New_grating

Cmd$=Cmd$&Dum$

WHILE Cur_grating<>New_grating

CALL F2000send("ITI1 300 WAVE TURN",1)

WAIT 1.5

Cur_grating=(Cur_grating MOD 3)+1

END WHILE

END IF

END IF

IF Curr_wave>Wavelen THEN ! eliminate backlash

Cmd$=Cmd$&VAL$(Wavelen-40)&" WAVE 

END IF

Cur_wave=Wavelen

In the original FOA-2000, the monochromator only covers the range from 800 to 1600 nm. For this range, only two cutoff filters were needed, one to cover 800 to 1000 nm, and another to cover 1000 to 1600 nm. The value at which the filter was switched was denoted in the software as a parameter named Cutoff. The NRL system requires five cutoff filters, so we need an array to pass the values of the wavelengths at which the cutoff filters should be switched. For this we use an array called Filter(*), which is passed to this subroutine by the Sysdata COM block. This was used in the old software, but in an early version of the FOA-2000.
the file that we are using a monochromator with cutoff filters, which is necessary in order to set the Filter_flag to 2 (which in turn flags the program to determine which value of Filter(*) to use).

The cut-on wavelengths for the cutoff filters are as follows:

- Filter(1) = 500 nm
- Filter(2) = 900 nm
- Filter(3) = 1525 nm
- Filter(4) = 2175 nm
- Filter(5) = 3150 nm

If Filter_flag=2, we are using the monochromator with cutoff filters installed in SEVERAL positions of the filter wheel, so decide which one to use. If Filter_flag=0, we are not using the cutoff filters.

IF Filter_flag=2 THEN
    FOR Index=11 TO 0 STEP -1
        IF Filters(Index)<#Wavelen THEN GOTO Change
    NEXT Index
END IF

Change: Cmd$=Cmd$&VAL$(Index)&" FILTER 

IF Wavelen<Less_switch THEN
    Cmd$=Cmd$&"GERMAIN" !Less than switch so use Germainium
ELSE
    Cmd$=Cmd$&"INS8" !Otherwise use Indium-Antimonide
END IF
CALL F2000send(Cmd$,1)

Done: SUBEXIT

Filter_err: BEEP

DISP "NEXTWAVE -- Wavelength "&VAL$(Wavelen)&" is not available on the filter wheel."

Done: GOTO Dead1

CALL Cleardisplay

SUB Clearup

CALL Cleardisplay

PRINT TABXY(15,8);"Please be patient, this might take a moment."
CALL F2000send("LED SPOT-OUT TARGET-OUT ILLUMIN VOUT GERMAIN")
CALL EggS205
CALL ES205com("A2 1")
CALL Setscale(.1,1)
BEEP
PRINT TABXY(22,14);"EG&G Lock-in cleared and reset!"
CALL F2000send("ALIGN")
WAIT 3
CALL Cleardisplay

SUBEND

SUBEND
This module contains the primary code to run a DMA measurement. It differs from a spectral attenuation measurement in that it allows a number of wavelength scans to be performed on a long length of fiber before cutback. NA Restrictors are requested for each run. After cutback, Restrictors are requested in the same order as used originally. Data is stored in two arrays, Dmarundata, for measurements before, and Dmarefdata, for measurements after cutback.

Set up parameters.

Dmarundata(0,0)=Numsteps
Dmarefdata(0,0)=Numsteps
Dmarundata(1,0)=Fiber_len
Dmarefdata(1,0)=Fiber_len
Dma_id$=Fiber_id$" &Log_time$

Once again: OFF KEY

Now the actual measurement loop.

PRINT TABXY(1,6);"Long fiber wavelength scan in progress using Restrictor ";Restr_no

FOR Wavecount=1 TO Numsteps
CALL Nextwave(Wavelength(Wavecount))
CALL Setfocus(Wavelength(Wavecount))
Measurement=FNVolmeter(.01)
Dmarundata(Wavecount,Run_no)=Measurement
NEXT Wavecount

CALL F2000send("O OUT-X 0 OUT-Y 0 OUT-Z") !In case of manual adjustment

Align2=FNVolmeter(.05)

CALL F2000send("LAMP LAMP-ON")

IF Align_change>1 THEN !More than 1% signal change
BEEP
CALL Cleardisplay
PRINT TABXY(1,10);"LED alignment signal changed."
9630  PRINT TABXY(1,14);"To PROCEED with the test, press f5. To EXIT th
9632  e test, press f8."
9634  ON KEY 1 LABEL "RE-RUN" GOTO Once_again
9636  ON KEY 5 LABEL "PROCEED" GOTO Choose_another
9638 Snoozed: GOTO Snoozed
9640  ELSE
9642  PRINT TABXY(1,8);"Alignment okay; test proceeding."
9644  END IF
9646 |
9648 Choose_another: OFF KEY
9650  PRINT TABXY(1,10);"To run another wavelength scan with another NA res
9652  trictor, press f1."
9654  ON KEY 1 LABEL "CHANGE RESTRCTR" GOTO Next_restr
9656  ON KEY 5 LABEL "GO TO CUTBACK" GOTO Cutback
9658  ON KEY 8 LABEL "EXIT" GOTO Done
9660 Catch_here: GOTO Catch_here
9662 |
9664 Cutback: OFF KEY
9666 Totalruns=Run_no  !Total number of runs (one per restrictor)
9668 Dmarundata(2,0)=Totalruns  !Store total number of runs here
9670 Dmarefdata(2,0)=Totalruns
9672 CALL Fiberload(" PLEASE CUT BACK THE FIBER")
9674 CALL Outalign
9676 |
9678  ! Now take measurements on the "ref" (i.e. short, cutback) fiber.
9680 FOR Run_no=1 TO Totalruns
9682  BEEP
9684  PRINT TABXY(1,10);"Please insert Restrictor ",Dmarefdata(0,Run_no)," and press f5 when ready."
9686  ON KEY 5 LABEL "PROCEED" GOTO And_again
9688 Hang_on_here: GOTO Hang_on_here
9690 |
9692 And_again: OFF KEY  !Set up bench; do it all in case of manual adjstmnt
9694 CALL F2000send("O IN-X 0 IN-Y 0 IN-Z")  !Make sure it's at 0 location
9696 CALL F2000send("O OUT-X 0 OUT-Y 0 OUT-Z")
9698 CALL F2000send("LED LED-ON CHOP-ON SPOT-OUT XMIT")
9700 CALL F2000send("TARGET-OUT VOUT GERMAIN")
9702 Align1=FNVoltmeter(.05)
9704 CALL F2000send("LAMP LAMP-ON")
9706 ! Now the actual measurement loop.
9708 !
9710 PRINT TABXY(1,20);"Cutback fiber wavelength scan in progress using Re
9712  !range ",Dmarundata(0,Run_no)
9714 FOR Wavecount=1 TO Numsteps
9716  CALL Nextwave(Wavelength(Wavecount))
9718  CALL Setfocus(Wavelength(Wavecount))
9720 Measurement=FNVoltmeter(.01)
9722  Umarefdata(Wavecount,Run_no)=Measurement
9724 NEXT Wavecount
9726 |
9728 CALL F2000send("O IN-X 0 IN-Y 0 IN-Z")  !Go back to alignment position
9730 CALL F2000send("O OUT-X 0 OUT-Y 0 OUT-Z")
9732 CALL F2000send("GERMAIN LED LED-ON")  !Prep for LED voltage reading
9734 ! Check signal integrity.
9736 Align_change=100*(Align1-Align2)/Align1
9738 IF Align_change>1 THEN !More than 1% signal change
9740 CALL Cleardisplay
9742 PRINT TABXY(1,10);
9744 PRINT USING "36A,MDD.D,9A";"The LED alignment signal changed by",A
e test, press F8.

9750 ON KEY 1 LABEL "RE-RUN" GOTO And_again
9752 ON KEY 5 LABEL "PROCEED" GOTO On_dasher
9754 ON KEY 8 LABEL "EXIT" GOTO Done
9756 Sleeper: GOTO Sleeper
9758 ELSE
9760 END IF
9762 Next Run_no
9764 On_dasher: OFF KEY
9766 NEXT Run_no
9768 !
9770 Done: OFF KEY
9772 CALL Cleardisplay
9774 LOCAL @Fo2000
9776 SUBEND
9778 !
9780 !
9782 SUB Dmacomp
9784 !----------------------------------------------------------------------------------------
9786 ! DIFFERENTIAL MODAL ATTENUATION COMPUTE MODULE
9788 !--------------------------------------------------------------------------------------------
9790 ! This module computes the fiber spectral attenuation for the different NA ranges used for the test.
9792 !--------------------------------------------------------------------------------------------
9794 !
9796 COM /Wavelength/ Wavelength(*),Numsteps
9798 COM /Dmatdata/ Dmarundata(*),Dmarefdata(*),Dmaattendata(*),Dma_id$
9800 !
9802 INTEGER I,J,Run_no,Totalruns
9804 REAL Steps_runs
9806 !
9808 Numsteps=Dmarundata(0,0)
9810 Fiber_len=Dmarundata(1,0)
9812 Totalruns=Dmarundata(2,0)
9814 ! Since only the (0,0) slot is open in the Dmaattendata array, parse
9816 ! the number of wavelength steps (up to 350) and the number of DMA runs
9820 ! with different Restrictors (up to 11) into the integer and fractional
9822 ! parts of a single variable called "Steps_runs".
9824 Steps_runs=Numsteps+Totalruns/100
9826 Dmaattendata(0,0)=Steps_runs
9828 !
9830 FOR I=1 TO Numsteps
9832 Dmaattendata(I,0)=Wavelength(I)
9834 FOR J=1 TO Totalruns
9836 Dmaattendata(0,J)=Dmarundata(I,J)
9838 Dmaattendata(I,J)=10*LGT(Dmarefdata(I,J)/Dmarundata(I,J))
9840 Dmaattendata(I,J)=Dmaattendata(I,J)/Fiber_len
9842 NEXT J
9844 NEXT I
9846 !
9848 SUBEND
9850 !
9852 !
9854 DEF FNGetrestrtctor$(Plot$)
9856 !----------------------------------------------------------------------------------------
9858 ! CHOOSE NA RESTRICTOR MODULE
9860 !----------------------------------------------------------------------
9862 ! This module is called before a DMA measurement to ask the user which NA Restrictor he desires to use for the test. It also pauses to allow
9864 ! the Restrictor to be put in the holder next to the cut-off filter wheel.
9866 ! After the test, or when reviewing recalled data, the module is called
9868 ! again to determine which column of data (one corresponding to each
9870 ! Restrictor) is used when the spectral attenuation module...
INTEGER Indexi, IndexJ, Restr_no, Totalruns

DIM Restr$(11)[17]

CALL Cleardisplay

! See if this is the first run, if so goto ask for a new Restrictor.
Totalruns=FRAC(Dmaattendata(0,0))*100
IF Totalruns=0 THEN GOTO New_restr

! Reprint:
PRINT TABXY(1,4);"NA Restrictor values in the present data set are:
FOR Indexi=1 TO Totalruns
PRINT TABXY(47+Indexi*3,4);Dmaattendata(0,Indexi)
NEXT Indexi

! Ask operator which restrictor to use.
New_restr:
OFF KEY
Restr$(0)="0 Full NA = .24"
Restr$(1)="1 NA = .04"
Restr$(2)="2 NA = .08"
Restr$(3)="3 NA = .10"
Restr$(4)="4 NA = .13"
Restr$(5)="5 NA = .15"
Restr$(6)="6 NA = .18"
Restr$(7)="7 NA = .20"
Restr$(8)="8 .04<NA<.08"
Restr$(9)="9 .08<NA<.13"
Restr$(10)="10 .11<NA<.17"
Restr$(11)="11 .14<NA<.21"
PRINT TABXY(19,6);" Restr.# NA Range ",
FOR Indexj=0 TO 11
PRINT TABXY(25,Indexj+7);Restr$(Indexj)
NEXT Indexj
PRINT BEEP
Restr_no=FNGetint("Enter the restrictor # to use: ",0,11)
CALL Cleardisplay

! The next condition being met means we're preparing to plot data;
in that case, go down and return to Dmaplotprep. Else load Restrictor.
FOR Indexi=1 TO LEN(Plot$)
IF Plot$="PLOT" THEN GOTO Headout
NEXT Indexi

Got_no:
IF Restr_no=0 THEN
PRINT TABXY(1,15);"No NA Restrictor was specified."
PRINT TABXY(1,18);"A straight Spectral Attenuation measurement will be performed."
ELSE
PRINT TABXY(1,15);
PRINT USING "30A,0D,25A";"Please insert NA Restrictor ",Restr_no," and press F5 when ready."
END IF
ON KEY I LABEL "CHANGE RESTRCTR" GOTO Kleenscreen
ON KEY 5 LABEL "PROCEED" GOTO Headout
! Head back to Dmarun
Prang: GOTO Prang
Kleenscreen: I
OFF KEY
OUTPUT KBD USING ";K","K"
GOTO Reprint
DETERMINE DATA SOURCE MODULE

This routine is called before each fiber test is performed, to determine where the data for the output plot is to come from. It allows a user to review data from a previous day (computer turned off overnight), data presently in the memory (earlier the same day), or run a new test.

To access data from an archived file, press RETRIEVE.
To review data presently in memory, press EXISTING DATA.
To begin a new Far Field measurement, press NEW TEST.

ON KEY 1 LABEL "RETRIEVE" GOTO Pullit
ON KEY 3 LABEL "EXISTING DATA" GOTO Existing
ON KEY 5 LABEL "NEW TEST" GOTO Newtest

Freeze: GOTO Freeze

Pullit: OFF KEY
CALL Retrieve
Source_flag=2
GOTO Scram

Existing: OFF KEY
Source_flag=1
GOTO Scram

New_test: OFF KEY
Source_flag=0
GOTO Scram

RETURN Source_flag

SUB Dmaplotprep
PREPARE DNA DATA FOR PLOTTING

COM /Specattdata/ Specattdata(*),Specatt_id$
COM /Dmadata/ Dmadata(*),Dmarefdata(*),Dmaattendata(*),Dma_Id$
INTEGER Indexl,Indexj,Indexk,Numsteps,Totalruns,Restr_no,Restri
d
DIM Restrictor$[30]

Numsteps=INT(Dmaattendata(0,0)) !Integer part is # wave steps
Totalruns=FINT(Dmaattendata(0,0))*100 !Fract part is # NA restr runs

Query:
Print Restrictor values in the data set; print NA ranges for each value.
query the user as to which set to plot; extract Restrictor number from string (returned from FNGetrestrictor); search data for desired column.
Restrictor$=FNGetrestrictor$("PLOT") !Query
String_len=LEN(Restrictor$)
Restrictor$="1 Restr # &Restrictor$[1-2]&; &Restrictor$[6-String_len]
Restr_no=VAL(Restrictor$[12:21]) !Extract Restr_no from string
10134 NEXT Run_no
10136 PRINT TABXY(1,12);"No match for this Restrictor # was found in the data. Please try again. (WAIT)"
10138 WAIT 4
10140 OUTPUT KBD USING ";K";"K" !Clear alpha's only
10142 GOTO Query
10144 
10146 Found_column: !Load the appropriate DMA data in the Specattdata array.
10148 Specatt_id$=Dma_id$&Restricter$
10150 Specattdata(0,0)=Dmarundata(0,0) !Transfer number of points
10152 Specattdata(0,1)=Dmarundata(1,0) !Transfer fiber length
10154 FOR Index=1 TO Numsteps-1
10156 Specattdata(Index,0)=Dmaattdata(Index,0) !Load wavelengths first
10158 Specattdata(Index,1)=Dmaattdata(Index,Run_no) !Now the data
10160 NEXT Index
10162 !Spectral attenuation routines now may be used to list and plot data.
10164 SUBEND
10166 !
10168 !
10170 SUB Ffsmooth(Data$)
10172 !*************************************************************************************************************
10174 ! SMOOTH FAR FIELD DATA
10176 !*************************************************************************************************************
10178 !This routine is provided to offer the user the option of smoothing the far field data by a variable pointwise number. Smoothing is generally desirable owing to the spike-generating tendency of the differentiation process used to derive the far field scan values.
10180 !
10182 !
10184 !
10186 !
10188 COM /Fftempdata/ Ffrawdata(*),Ffdiffdata(*),Ffsmoothdata(*)
10190 !
10192 INTEGER Smoothpts,I,J,Numpoints
10194 !
10196 !First ask for the number of points to use in the smoothing operation.
10198 !Set an upper limit of a 25 point smooth (changeable if necessary).
10200 !
10202 CALL Cleardisplay
10204 PRINT TABXY(1,11);" 
10206 Smoothpts=FNGetint("Enter the number of points to use in the smoothing procedure:",0,25)
10208 IF Smoothpts=0 THEN Smoothpts=1 !Smooth by 0 pts really means 1
10210 CALL Cleardisplay
10212 !
10214 !Next determine which data set to smooth (RAW or ROUGH), and smooth it.
10216 !
10218 IF Data$="RAW DATA" THEN
10220 Numpoints=Ffrawdata(0,0)-Smoothpts+1
10222 Ffsmoothdata(0,0)=Numpoints
10224 FOR I=1 TO Numpoints
10226 Total=0
10228 FOR J=I TO Smoothpts+I-1
10230 Total=Total+Ffrawdata(J,1)
10232 NEXT J
10234 Ffsmoothdata(I,0)=Ffrawdata(I,0)
10236 Ffsmoothdata(I,1)=Total/Smoothpts
10238 NEXT I
10240 END IF
10242 !
10244 IF Data$="DIFF" THEN
10246 Numpoints=Ffdiffdata(0,0)-Smoothpts+1
10248 Ffsmoothdata(0,0)=Numpoints
10250 FOR I=1 TO Numpoints
10252 Total=0
10254 NEXT I
10256 FOR J=I TO Smoothpts+I-1
10258 Total=Total+Ffdiffdata(J,1)
10260 NEXT J
10262 Ffsmoothdata(I,0)=Ffdiffdata(I,0)
10264 Ffsmoothdata(I,1)=Total/Smoothpts
10266 NEXT I
10268 END IF
10270 !
Ffsmoothdata(I,1)='Total'/Smoothpts

IF Data$="SMOOTH" THEN
   Num_points=Ffsmoothdata(0,0)-Smoothpts+1
   Ffsmoothdata(0,0)=Num_points
   FOR I=1 TO Num_points
      Total=0
      FOR J=1 TO Smoothpts+I-1
         Total=Total+Ffsmoothdata(J,1)
      NEXT J
      Ffsmoothdata(I,1)=Total/Smoothpts
   NEXT I
END IF

IF smoothpts>1 THEN
   IF Data$="SMOOTH" THEN
      PRINT TABXY(1,4);"Data further smoothed ";
      PRINT USING "7A,DD,15A";"using a ",Smoothpts," point average."
   ELSE
      PRINT TABXY(1,4);"Data smoothed using"
      PRINT USING "7A,DD,15A";"a ",Smoothpts," point average."
   END IF
END IF

SUBEND

SUB Cleardata

INTEGER I,J

IF Specattdata(0,0)<0 THEN
   FOR I=0 TO Specattdata(0,0)
      FOR J=0 TO I
         Specattdata(I,J)=0
      NEXT J
   NEXT I
   Specatt_id$=" "
   Clear$="CLEAR"
END IF

IF Darundata(0,0)<0 THEN
   FOR I=0 TO Darundata(0,0)
      FOR J=0 TO Darundata(2,I)
         Darundata(I,J)=0
      NEXT J
   NEXT I
END IF

CLEAR DATA MODULE

This routine can be called to effectively clear all data from memory. It simply sets all parameters equal to 0.
IF Field(0,0)<>0 THEN
FOR I=0 TO Field(0,0)
    FOR J=0 TO 1
        Field(I,J)=0
        Ffrawdata(I,J)=0
        Ffdiffdata(I,J)=0
        Ffsmoothdata(I,J)=0
    NEXT J
NEXT I
Field_id$=""
Clear$="CLEAR"
END IF
CALL Cleardisplay
IF Clear$="CLEAR" THEN
    PRINT TABXY(20,10);"DATA HAS BEEN CLEARED."
ELSE
    PRINT TABXY(20,10);"NO DATA TO CLEAR."
END IF
WAIT 2
CALL Cleardisplay
SUBEND

SUB Proglist
PRINT PROGRAM LISTING OR PROGRAM CONTENTS
CALL Cleardisplay
PRINT TABXY(8,8);"To print a list of the subroutines contained in the master program, as well as their locations within the program, press f1."
PRINT TABXY(8,9);"To print the entire program, press f5."
PRINT TABXY(8,12);"Be forwarned that this take an hour or more."
ON KEY 1 LABEL "CONTENTS" GOTO Contents
ON KEY 8 LABEL "PROGRAM LISTING" GOTO Listit
Crashout: GOTO Crashout
Listit: OFF KEY
PRINT TABXY(20,16);"PRINTING PROGRAM LISTING"
PRINTER IS PRT
LIST
PRINTER IS CRT
OUTPUT KBD USING ";K";
SUBEXIT

Contents: OFF KEY
PRINT TABXY(20,16);"PRINTING PROGRAM CONTENTS"
PRINTER IS PRT
ON KEY 8 LABEL "PROGRAM LISTING" GOTO Listit
ON KEY 8 LABEL "PROGRAM LISTING" GOTO Listit
Crashout: GOTO Crashout

PRINT "ROUTINE NAME LINE NUMBER"
PRINT "-------------------------------------------------------------------"
PRINT "Mainprog 10"
PRINT "Sysihit 334"
PRINT "Systemdata 719"
SUB FIBERTEST1 (OPTIONAL Source_flag)

FIBERTEST SUBPROGRAM NO. 1 -- SPECTRAL ATTENUATION

DIM Flags$[10]

IF NPAR > 0 THEN

IF Source_flag = 0 THEN GOTO New_test!

IF Source_flag = 1 THEN GOTO Scale

IF Source_flag = 2 THEN GOTO Plot_spec

END IF

New_test!: 

CALL Fiberload(" Please load the test fiber.")

CALL Fiberident

CALL Askalign

CALL Logtime

CALL Specrun("OVERFILL")

CALL F2000end("GERMAIN")

CALL Fiberload("PLEASE CUT BACK THE FIBER")

CALL Outalign

CALL Specref("OVERFILL")

CALL Specatcomp

Scale: 

PRINT TABXY(20,16);"Select the desired range for the plot."

BEEP

Flags$=""

ON KEY 1 LABEL " dB/km" GOTO Kilo_db

ON KEY 2 LABEL " dB/100m" GOTO Hundred_db

ON KEY 3 LABEL " dB/10m" GOTO Ten_db

ON KEY 4 LABEL " dB/m" GOTO Db_per_m

OUT_to_lunch: GOTO OUT_to_lunch

Kilo_db:

Flags$="KILO"

GOTO Got_factor

Hundred_db: 

Flags$="HUNDRED"

GOTO Got_factor

Ten_db: 

Flags$="TEN"

GOTO Got_factor

Db_per_m: 

Flags$="METER"

GOTO Got_factor

Plot_spec: 

Flags$="2"

CALL Specatcomp

Got_factor:!

OFF KEY

DISP

OUTPUT KBD USING ";";

CALL Specatplot(Flags$)

Test flag values returned from Specatplot for where to go from here:

IF Flags$="RESCALE" THEN GOTO Scale

Rescale plot and do again

IF Flags$="LISTING" THEN GOTO Print_list

Print hard copy listing

IF Flags$="STORE" THEN GOTO Stor3it

Archive data

IF Flags$="QUIT" THEN GOTO Done

None of the above

GOTO Done

Print_list:!

OFF KEY

DISP
 SUB Fibertest2(OPTIONAL Source_flag)

 IF NPAR>0 THEN  
    IF Source_flag=0 THEN GOTO New_test  
    IF Source_flag=1 THEN GOTO Plot_prep  
    IF Source_flag=2 THEN GOTO Plot_prep  
 END IF  

 New_test: 
 CALL Fiberload("Please load the test fiber.") CALL Fiberident CALL Askalign CALL Logtime CALL Dmarun CALL Dmacomp Plot_prep:  
 CALL Dmaplotprep  

 Prepare scale information for Specattplot: 

 Scale:  
 PRINT TABXY(20,16);"Select the desired range for the plot." BEEP Flags$=""  
 ON KEY 1 LABEL " dB/km" GOTO Kilo_db  
 ON KEY 2 LABEL " dB/100m" GOTO Hundred_db  
 ON KEY 3 LABEL " dB/10m" GOTO Ten_db  
 ON KEY 4 LABEL " dB/m" GOTO Db_per_m  
 OUT_to_lunch: GOTO Out_to_lunch  
 Kilo_db:  
 Flags$="KILO" GOTO Got_factor  
 Hundred_db:  
 Flags$="HUNDRED" GOTO Got_factor  
 Ten_db:  
 Flags$="TEN" GOTO Got_factor  
 Db_per_m:  
 Flags$="METER" GOTO Got_factor  
 Got_factor:  
 OFF KEY  
 CALL Cleardisplay  
 CALL Specattplot(Flags$,,0,,"DIFFERENTIAL MODAL ATTENUATION")
10796 IF Flags$="QUIT" THEN GOTO Done  

10801 OUTPUT KBD USING ",K","K"  

10802 CALL Specatlist("PRINT ",Flags$,"DIFFERENTIAL MODAL ATTENUATION

Restrictor #:"&VAL$(Restr_no))

10803 GOTO Done

10805 CALL Archive

10806 CALL Cleardisplay

10808 CALL Archive

10810 SUBEND

10812 |

10813 SUB Fibertest3(Optional Source_flag)

10814 |*************************************************************|

10815 ! FIBERTEST SUBPROGRAM NO. 3 -- FAR FIELD

10816 |*************************************************************|

10817 |

10818 ! First test whether or not to run a new test, or go directly to plot.

10819 |

10820 IF NPAR>0 THEN  

10821 IF Source_flag<>0 THEN GOTO Plotit  

10822 END IF

10823 |

10824 CALL Fiberload("Please load the test fiber.")

10825 CALL Fiberident

10826 BEEP

10827 Ftwave=FNGetffwave

10829 CALL Askalign

10829 CALL Ffieldvals("-.35 TO .35 STEP .0075")

10830 CALL Logtime

10831 CALL Ffieldrun(Ftwave)

10832 CALL Fnormalize("RAW DATA")

10833 Plotit: |

10834 CALL Ffieldplot("RAW DATA","Far-Field Raw Data (before differentiation)"

10835 CALL Fdiff

10836 CALL Fcorrect

10837 Print_flag$="DIFF"

10838 Normalize_it:

10839 CALL Fnormalize(Print_flag$)

10840 CALL Ffieldplot(Print_flag$,"FAR FIELD PATTERN")

10841 IF Print_flag$="SMOOTH" THEN GOTO Normalize_it

10842 SUBEND

10843 |

10845 SUB Fibertest4(Optional Source_flag)

10846 |*************************************************************|

10847 ! Fibertest4: FARFIELD WITH PINHOLE  

10848 |

10849 |

10850 |*************************************************************|

10851 CALL Cleardisplay
10855 Fpintest: OFF KEY
10856 CALL Fiberload("Please load the test fiber.")
10857 CALL Fiberident
10858 CALL Askalign
10859 CALL Fieldvals("-.35 TO .35 STEP .0075")
10860 CALL Fieldrun(FNGetffwave,"PINHOLE")
10861 CALL Fcorrect
10862 CALL Fnormalize("RAW")
10863 CALL Fieldplot("RAW","Far Field Pattern (using pinhole)")
10864 SUBEND
10865 |
10866 |
10867 SUB Fibertest5(OPTIONAL Source_flag)
10868 |+-------------------------------------------------------------------------
10869 | FIBERTEST SUBPROGRAM NO. 5 -- NEAR FIELD
10870 |--------------------------------------------------------------------------
10871 | This routine is presently inactive. To include the nearfield test as
10872 | a test option, see NRL IR System1 Operating Manual for basic needs.
10873 |
10874 CALL Cleardisplay
10875 PRINT TABXY(12,10):"It said this test is INACTIVE. Can't you read?"
10876 PRINT TABXY(25,12):"(Don't touch that dial!)
10877 WAIT 4
10878 SUBEXIT
10879 |
10880 | The real program begins here: Please load the test fiber.")
10881 CALL Fiberload("Please load the test fiber.")
10882 CALL Fiberident
10883 CALL Askalign
10884 CALL Nfieldvals("-35 to -20.5 STEP .5,-20 TO 20 STEP 2,20.5 TO 35 STEP 5")
10885 CALL Logtime
10886 CALL Nfieldrun
10887 CALL Nfieldplot
10888 SUBEND
10889 |
10890 |
10891 SUB Fibertest6
10892 |+-------------------------------------------------------------------------
10893 | FIBERTEST SUBPROGRAM NO. 6 -- FIBER LOADING & IDENTIFICATION
10894 |--------------------------------------------------------------------------
10895 CALL Fiberload("Please load the test fiber.")
10896 CALL Fiberident
10897 SUBEND
10898 |
10899 |
10900 |
10901 SUB Fibertest6
10902 |+-------------------------------------------------------------------------
10903 | FIBERTEST SUBPROGRAM NO. 6 -- FIBER LOADING & IDENTIFICATION
10904 |--------------------------------------------------------------------------
10905 CALL Fiberload("Please load the test fiber.")
10906 CALL Fiberident
10907 SUBEND

ROUTINE NAME | LINE NUMBER
-------------|----------------
Mainprog      | 10
Sysinit      | 334
Systemdata   | 719
Timiset      | 1726
FNTimedata$  | 1900
Logtime      | 1996
Archive      | 2022
Retrieve     | 2300
Zcenter      | 2548
Rundisplay   | 2582
Cleardisplay | 2608
F2000and     | 2630
Preset       | 2736
EggS205comm  | 2930
FNVoltmeter  | 3016